Fast Algorithm of Short-time DCT for Low Resolution Signal Processing

Vitaly Kober
Department of Computer Science, Division of Applied Physics, CICESE,
Carretera Ensenada-Tijuana No. 3918, Ensenada, B.C. 22860, Mexico

Keywords: Discrete Cosine Transform, Fast Algorithm.

Abstract: A fast algorithm for computing the discrete cosine transform (DCT) in a window running on a signal with a step higher than one is proposed. The algorithm is based on a second-order recursive relation between DCT spectra computed in windows which are equally spaced with a given distance. The computational complexity of the proposed algorithm is compared with that of common fast and running DCT algorithms. A fast inverse DCT transform is also presented.

1 INTRODUCTION

The discrete cosine transform (DCT) is widely used in many signal processing applications such as adaptive filtering, video signal processing, feature extraction, and data compression. This is because the DCT performs close to the Karhunen-Loeve transform for the first-order Markov stationary data, when the correlation coefficient is close to unity (Jain, 1979).

In many applications, time-varying signals have inherently infinite length. Since the signal properties (amplitudes, frequencies, and phases) usually change with time, a single orthogonal transform is not sufficient to describe the entire signal. The concept of short-time signal processing with filtering in the domain of an orthogonal transform may be utilized (Oppenheim & Shafer, 1989). A particular case of the short-time processing is a local signal processing in the domain of an orthogonal transform calculated in a running window. The running orthogonal transform of a signal x_k can be defined as (Vitkus & Yaroslavsky, 1987)

$$X_s^{kp} = \sum_{n=-N_1}^{N_2} x_{kp+n} \psi(n,s), \qquad (1)$$

where $\{\psi(n,s)\}$ represents the basis of orthogonal transform functions, k is integer value, p is the step of the running window, and X_s^{kp} displays the orthogonal transform coefficients of the signal around time kp. A block-diagram of this processing is illustrated in Fig. 1.

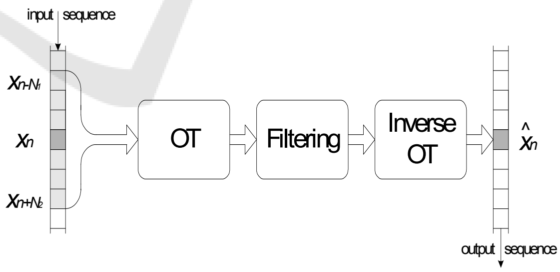

Figure 1: A block-diagram of local signal processing in the domain of an orthogonal transform in a running window.

The choice of orthogonal transform for running processing depends on many factors. The DCT is one the most appropriate transforms with respect to the accuracy of power spectrum estimation from the observed data that are required for local filtering, the filter design, and computational complexity of the filter implementation. The kernel of the DCT for the order N is defined as

$$DCT_N = \left\{ k_s \cos\left(\pi \frac{s(n+1/2)}{N} \right) \right\} \qquad (2)$$

where $n, s=0,\ldots, N-1,$ $k_s = \begin{cases} 1/\sqrt{2}, & \text{if } s=0 \\ 1, & \text{otherwise} \end{cases}$

For clarity, the normalization factor $\sqrt{2/N}$ can be

Kober V..
Fast Algorithm of Short-time DCT for Low Resolution Signal Processing.
DOI: 10.5220/0004017800710074
In *Proceedings of the International Conference on Signal Processing and Multimedia Applications and Wireless Information Networks and Systems* (SIGMAP-2012), pages 71-74
ISBN: 978-989-8565-25-9

neglected until the inverse transform. The computation of the DCT at each position of a moving window is an intensive task. When the window moves with a step of one, the shift properties of discrete sinusoidal transforms may be used for updating the transform coefficients. The shift properties of the first-order were derived (Yip and Rao, 1987). However, this approach is not very efficient with respect to computational complexity. The second-order recursive expressions and rapid algorithms for the computation of the running DCT with a step of one were suggested (Macias and Exposito, 1998); (Kober and Cristobal, 1999); (Xi and Chiraro, 2000); (Kober, 2004). In real applications, it is often necessary to process a signal with a high rate and a low resolution; for instance, in order to obtain preliminary results for further data analysis. In this case, signal processing in the domain of the sliding DCT with a given arbitrary step (higher that one) is appropriate technique.

In this paper, a fast recursive algorithm for computing the DCT in a window running on a signal with an arbitrary step with less computation than that of known recursive and fast DCT algorithms is proposed. Moreover, intervals for the window step in which the proposed algorithm is more effective than the fast DCT algorithms are calculated. A fast inverse DCT transform is also presented.

2 RECURSIVE ALGORITHM FOR COMPUTING RUNNING DCT WITH ARBITRARY STEP

The running DCT with a step of p can be defined as follows:

$$X_s^{kp} = \sum_{n=-N_1}^{N_2} x_{kp+n} \cos\left(\pi \frac{\left(n+N_1+1/2\right)s}{N}\right), \qquad (3)$$

where $\{x_k;\ k = \ldots,\ -N_1,\ -N_1+1, \ldots,\ 0,\ 1, \ldots,\ N_2,\ N_2+1 \ldots\}$ is an infinite-length signal, $\left\{X_s^{kp}, s = 0,1,\ldots N-1\right\}$ are the transform coefficients around time kp, $N = N_1 + N_2 + 1$ is the length of the running window. Here N is an arbitrary integer value. For clarity, the normalization factor and specific scaling of the X_0^{kp} component are neglected. The coefficients of the DCT can be obtained as $\{C_0^{kp} = X_0^{kp}/\sqrt{2}$; $C_s^{kp} = X_s^{kp}$, $s = 1,\ldots N-1\}$. A recursive relationship between three consecutive running spectra X_s^{k-2}, X_s^{k-1} and X_s^k is given by (Kober, 2004)

$$X_s^k = a_s X_s^{k-1} - X_s^{k-2} + f_s^k, \qquad (4)$$

where $f_s^k = \left(x_{k-N_1-2} - x_{k-N_1-1} + (-1)^s \left(x_{k+N_2} - x_{k+N_2-1}\right)\right)$ x $\cos(\alpha_s/2)$, $a_s = 2\cos(\alpha_s)$, and $\alpha_s = \dfrac{\pi s}{N}$.

This is a second-order liner inhomogeneous difference equation defined on k. The roots of the corresponding characteristic equation (Kelley & Peterson, 1991; Mickens, 1990) are equal to $\lambda_s = \exp(j\alpha_s)$ and $\gamma_s = \exp(-j\alpha_s)$. After some manipulations, the general solution of the difference equation (4) can be expressed as

$$X_s^k = \begin{cases} \dfrac{1}{k_1}\left(kX_s^{k_1} + (k-k_1)\left(\displaystyle\sum_{m=1}^{k_1-1} mf_s^{k_0+m+1} - X_s^{k_0}\right)\right) + \\ \displaystyle\sum_{m=1}^{k-k_1} mf_s^{k-m+1},\ s=0 \\ \dfrac{X_s^{k_1}\sin(\alpha_s k) + \left(\displaystyle\sum_{m=1}^{k_1-1} f_s^{k_0+m+1}\sin(\alpha_s m) - X_s^{k_0}\right)}{\sin(\alpha_s)\sin(\alpha_s k_1)} \text{x} \\ \sin(\alpha_s(k-k_1)) + \displaystyle\sum_{m=1}^{k-k_1} f_s^{k-m+1}\sin(\alpha_s m)/\sin(\alpha_s), \\ s = 1,\ldots N-1 \end{cases}, \qquad (5)$$

where $X_s^{k_0}$ and $X_s^{k_1}$ are initial DCT spectra computed at arbitrary window positions k_0 and k_1 ($k_0 < k_1$), respectively.

Since the window moves on a signal with a step of p, the local spectra are calculated in equally spaced window positions kp, where k is integer value, and $k > k_1$. In this case, a recursive relation between three spectra $X_s^{(k-2)p}$, $X_s^{(k-1)p}$, and X_s^{kp} can be obtained from equation (5) as follows:

$$X_s^{kp} = \begin{cases} 2X_s^{(k-1)p} - X_s^{(k-2)p} + \displaystyle\sum_{m=1}^{p-1}\left(f_s^{(k-2)p+m+1} + f_s^{kp-m+1}\right)m + \\ f_s^{(k-1)p+1}p,\ s=0 \\ 2X_s^{(k-1)p}\cos(\alpha_s p) - X_s^{(k-2)p} + \\ +\displaystyle\sum_{m=1}^{p-1}\left(f_s^{(k-2)p+m+1} + f_s^{kp-m+1}\right)\dfrac{\sin(\alpha_s m)}{\sin(\alpha_s)} + \\ f_s^{(k-1)p+1}\dfrac{\sin(\alpha_s p)}{\sin(\alpha_s)}, s = 1,\ldots N-1 \end{cases} \qquad (6)$$

We see that the computation of the DCT at the window position kp involves values of the input signal x_k as well as the DCT's coefficients computed in two previous positions $(k-1)p$ and $(k-2)p$ of the moving window. If $p=1$ the recursive equation is simplified to one given in (4).

The number of arithmetic operations required for computing the DCT with $p>1$ at a given window position is evaluated as follows: if N is odd, the complexity is $N(p+1)$ multiplication operations and $N(p+1)+5(p-1)$ addition operations; if N is even, from the property of symmetry of the functions $\left|\sin(\alpha_s m)/\sin(\alpha_s)\right| = \left|\sin(\alpha_{N-s} m)/\sin(\alpha_{N-s})\right|$ and $(-1)^s = (-1)^{N-s}, s = 1,...N-1, \quad m = 1,...p$, the computational complexity with respect to multiplication operations is reduced to $N(p+3)/2$. The total number of required arithmetic operations can be estimated as follows: $2N(p+1)+5(p-1)$ and $\frac{N}{2}(3p+5)+5(p-1)$ when N is odd and even, respectively.

Table 1: Comparison of algorithms for computing running DCT in terms of arithmetic operations flops (real adds and mults), N=256.

p-step of running window	Algorithms		
	Fast DCT	Recursive DCT	Proposed DCT
1	3708	1024	1024
2	3708	2048	1413
3	3708	3072	1802
4	3708	4096	2191
5	3708	5120	2580
6	3708	6144	2969
7	3708	7168	3358
8	3708	8192	3747
9	3708	9216	4136

The recursive running DCT algorithm (Rosendo et al., 1998); (Kober and Cristobal, 1999) requires approximately $4Np$ arithmetic operations for computing the DCT spectra in equally spaced window positions kp. One can observe that the proposed algorithm with $p>1$ is more effective than the use of the sliding recursive algorithm. Table 1 shows numerical results of computational complexity for the proposed, fast DCT (Johnson and Frigo, 2007); (Shao and Johnson, 2008), and running recursive algorithms (Rosendo et al., 1998); (Kober and Cristobal, 1999) for $N=256$ when p is varied. We see that the proposed algorithm yields better results when the window step p is less than a boundary value of 8. The boundary values for the step versus the window length N are provided in Table 2.

Table 2: Boundary values of the step p for the proposed algorithm versus N.

N	16	32	64	128	256	512	1024
p	3	4	6	7	8	9	10

The length of a moving window for the proposed algorithm may be an arbitrary integer value determined by characteristics of the signal to be processed.

The inverse algorithms for the sliding DCTs can be written as follows:

$$x_k = \frac{1}{N}\left(2\sum_{s=1}^{N-1} X_s^k \cos\left(\pi\frac{(N_1+1/2)s}{N}\right) + X_0^k\right), \qquad (7)$$

where $N=N_1+N_2+1$. The computational complexity is N multiplication operations and N addition operations. If x_k is the central pixel of the window, that is, $N_1=N_2$ and $N=2N_1+1$, then the inverse transform is given by

$$x_k = \frac{1}{N}\left(2\sum_{s=1}^{N_1}(-1)^s X_{2s}^k + X_0^k\right), \qquad (8)$$

So, in the computation only the spectral coefficients with even indices are involved. The computation requires one multiplication operation and N_1+1 addition operations.

3 CONCLUSIONS

A rapid recursive algorithm for computing the DCT in a window running on a signal with an arbitrary step was proposed. The algorithm is based on a recursive relationship between three consequent local DCT spectra which are computed in equally spaced signal windows. The computational complexity of the algorithm was compared with that of common fast and running DCT algorithms. Intervals for the window step in which the proposed algorithm is more effective than the fast DCT algorithms were calculated. The algorithm can be used for local signal processing with an arbitrary resolution in the short-time DCT domain.

REFERENCES

Jain, A. K., 1979. 'A sinusoidal family of unitary transforms', *IEEE Trans. Pattern Anal. Machine Intell.*, vol. PAMI-1, pp. 356-365.

Johnson, S. G. & Frigo, M. 2007. 'A modified split-radix FFT with fewer arithmetic operations', *IEEE Trans. Signal Process*., vol. 55, pp 111-119.

Kelley, W. G. & Peterson, A. C. 1991. *Difference equations. An introduction with Applications,* Academic, New York.

Kober, V. & Cristobal, G., 1999. 'Fast recursive algorithms for short-time discrete cosine transform', *Electronics Letters*, vol. 35, pp. 1236-1238.

Kober, V. 2004. 'Fast algorithms for the computation of sliding discrete sinusoidal transforms', *IEEE Trans. Signal Process*., vol. 52, pp. 1704-1710.

Mickens, R., 1990. *Difference equations*, Van Nostrand Reinhold, New York.

Oppenheim, A. V. & Shafer, R. W., 1989. *Discrete-time signal processing*, Prentice Hall, Englewood Cliffs NJ.

Rosendo Macias, J. A. & Gomez Exposito, A., 1998. 'Recursive formulation of short-time discrete trigonometric transforms', *IEEE Trans. Circuits. Syst. II*, vol. 45, pp. 525-527.

Shao, X. & Johnson, S. G. 2008. 'Type-II/III DCT/DST algorithms with reduced number of arithmetic operations', *Signal Process*., vol. 88, pp. 1553-1564.

Vitkus, R. Y. & Yaroslavsky, L. P., 1987. 'Recursive algorithms for local adaptive linear filtration', in Mathematical research, eds L.P. Yaroslavsky, A. Rosenfeld & Wilhelmi, Academy Verlag, Berlin.

Xi, J. & Chiraro, J. F., 2000. 'Computing running DCT's and DST's based on their second-order shift properties', *IEEE Trans. Circuits. Syst. I*, vol. 47, pp. 779-783.

Yip, P. & Rao, K. P., 1987. 'On the shift properties of DCT's and DST's', *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-35, pp. 404–406.