Discrete Orthogonal Transforms: A Review

P.B. Mohapatra

Scholar, CUTM,Paralakhemundi.
priyabrataballistics@gmail.com

S.S. Nayak

Professor in Physics, Centurion University of Technology & Management

ssnayak@cutm.ac.in

Abstract-Amongst the discrete orthogonal transforms, discrete cosine transform (DCT) and discrete sine transform (DST) are the most popular transforms and considered as the best substitute of the Karhunen-Loeve transform (KLT), not only for their near optimal performance but also for computational convenience. Therefore, the DCT and DST have been widely used in data compression, filtering and feature extraction applications. Several algorithms are reported in literature for efficient implementation of the DCT and DST in general – purpose computers, and also in dedicated VLSI. The VLSI systems yield high throughput of results by maximizing the processing concurrency, so that they provide less expensive and more suitable alternative to general – purpose computers, for real – time and on-line applications.

INTRODUCTION

In 1974, Ahmed *et al.* [1] proposed a real-valued discrete transform called the discrete cosine transform (DCT) and in 1976, A.K.Jain [19] proposed a real-valued discrete transform called the discrete sine transform (DST) which have emerged as the most popular substitute of the Karhunen-Loeve transform (KLT) in several speech and image signal processing applications [12, 38]. The KLT is known to be optimal with respect to the performance measures like the variance distribution [2], the mean-square criterion [32, 34] and the rate distortion function [31, 33] but there is no general algorithm for its fast computation. Compared with other orthogonal transforms such as the Walsh Hadamard transform (WHT), the discrete Fourier transform (DFT), the Haar transform (HT), and the Slant transform (ST), the performances of DCT and DST are found to be more close to that of the KLT.

DCT and DST

The DCT of a sequence { $x(n)$, $n = 0,1,2,\ldots, N-1$} is defined as [1]

$$X(k) = \frac{2}{N}\,\epsilon(k)\sum_{n=0}^{N-1} x(n)\cos\left[\frac{\pi(2n+1)k}{2N}\right] \quad (1)$$

The inverse discrete cosine transform (IDCT) is given by

$$x(n) = \frac{2}{N}\sum_{k=0}^{N-1} \epsilon(k)X(k)\cos\left[\frac{\pi(2n+1)k}{2N}\right] \quad (2)$$

for k =0,1,2,…., N-1 where $\epsilon(k) = \begin{bmatrix} 2^{-1/2} & \text{for } k=0 \\ 1 & \text{for } 1 \le k \le N-1 \end{bmatrix}$

The DST of a sequence { $x(n)$, n = 1,2,…..,N } is defined as [19]

$$X(k) = \frac{2}{N}\,\epsilon(k)\sum_{n=1}^{N} x(n)\sin\left[\frac{\pi(2n+1)k}{2N}\right] \qquad (3)$$

and the inverse discrete sine transform (IDST) is given by

$$x(n) = \sum_{K=1}^{N} \epsilon(k)X(k)\sin\left[\frac{\pi(2n-1)k}{2N}\right] \qquad (4)$$

for k = 1,2, ………..., N.

where $\epsilon(k) = \left[\begin{matrix} 2^{-1/2} & \text{for } k=N, \\ 1 & \text{otherwise} \end{matrix}\right.$

Since $\epsilon(k)$ effects only the amplitude of $X(N)$ component, we shall take $\epsilon(k)$ as unity with $X(N)$ scaled up by √2.

The DCT defined by equation (1) can be written in matrix form as

$$\mathbf{x} = \frac{2}{N}\,\mathbf{C}_N\,\mathbf{x} \qquad (5)$$

The DST defined by equation (3) can be written in matrix form as

$$\mathbf{x} = \frac{2}{N}\,\mathbf{S}_N\,\mathbf{x} \qquad (6)$$

Where **X** and **x** are the column vectors denoting the DCT/DST and the input data sequences, respectively, arranged in natural order. $\mathbf{C_N}$ denotes the DCT matrix of order N x N defined by equation (1) . $\mathbf{S}_N$ denotes the DST matrix of order N x N defined by equation (3). Without loss of generality , the normalizing factor 2/N is neglected in the rest of the thesis for convenience. Since DCT and DST are orthonormal , the forward transform can also be realized by taking transpose of the inverse transform.

The elements of the DCT matrix are given by

$$C_N(k,n) = \cos\left[\frac{\pi(2n+1)K}{2N}\right] \qquad (7) \qquad \text{for } k,n = 0,1,2,\ldots\ldots, N\text{-}1.$$

The elements of the DST matrix are given by

$$S_N(k,n) = \sin\left[\frac{\pi(2n-1)K}{2N}\right] \qquad (8) \qquad \text{for } k, n = 1,2,\ldots. N.$$

The orthogonal property of DCT can be expressed as

$$\mathbf{C}_N \mathbf{C}_N^T = \frac{N}{2}[\mathbf{I}] \tag{9}$$

where $\mathbf{C}_N^T$ is the transpose of $\mathbf{C}_N$.

The orthogonal property of DST can be expressed as

$$\mathbf{S}_N \mathbf{S}_N^T = \frac{N}{2}[\mathbf{I}] \tag{10}$$

Where $\mathbf{S}_N^T$ is the transpose of $\mathbf{S}_N$ and $[\mathbf{I}]$ is N x N identity matrix . It can be demonstrated that the basis sets of DCT and DST provide a good approximation to the eigenvectors of the normalized covariance matrix.

$$\boldsymbol{\Psi}=\begin{bmatrix} 1 & \varphi & \varphi^2 & . & . & . & \varphi^{N-1} \\ \varphi & \varphi^2 & \varphi^3 & . & . & . & \varphi^{N-2} \\ . & . & . & . & . & . & . \\ \varphi^{N-1} & \varphi^{N-2} & . & . & . & . & 1 \end{bmatrix}, \quad 0< \varphi <1 \tag{11}$$

Where φ is the one-step inter – element co- relation coefficient of a first order Markov process. The covariance matrix in the transform domain is given by

For DCT, $\Psi' = \mathbf{C}_N \, \Psi \, \mathbf{C}_N^{*T}$ (12)

For DST, $\Psi' = \mathbf{S}_N \, \Psi \, \mathbf{S}_N^{*T}$ (13)

Where $\mathbf{C}_N^{*T}$ is the complex conjugate $\mathbf{C}_N$ and $\mathbf{S}_N^{*T}$ is the complex conjugate of $\mathbf{S}_N$. From equations (12) and (13), it follows that Ψ'can be computed as a two-dimensional transform of $\boldsymbol{\Psi}$.

Several versions

Several versions of the DCT and DST such as even DCT(EDCT), even DST (EDST), odd DCT (ODCT), odd DST (ODST), symmetric DCT (SDCT), symmetric DST (SDST) and Hadamard DCT(HDCT) etc. have been proposed by researchers with a view to achieve optimality in performance and computational simplicity. Wang showed that there can be four different types of DCT and DST, to be used in varied situations. The following equations denote the different versions of the DCTs and DSTs.

$$[C_{N+1}^{I}] = \left(\frac{2}{N}\right)^{1/2} \left\{\epsilon(k)\varepsilon(n) \cos\left(\frac{kn\pi}{N}\right)\right\} \tag{14}$$

$k,n = 0,1,\dots, N$

$$[C_N^{II}] = \left(\frac{2}{N}\right)^{1/2} \left\{\epsilon(k) \cos\left[k\left(n+\frac{1}{2}\right)\frac{\pi}{N}\right]\right\} \quad (15)\ k,n = 0,1,\dots\dots,N\text{-}1$$

$$[C_N^{III}] = \left(\frac{2}{N}\right)^{1/2} \left\{\epsilon(n) \cos\left[n\left(k+\frac{1}{2}\right)\frac{\pi}{N}\right]\right\} \tag{16}$$

$k,n = 0,1\dots., N\text{-}1$

$$[C_N^{IV}] = \left(\frac{2}{N}\right)^{1/2} \left\{\cos\left(k+\frac{1}{2}\right)\left(n+\frac{1}{2}\right)\frac{\pi}{N}\right] \quad (17)$$

$k,n = 0,1,\ldots, N\text{-}1$

$$[S_{N-1}^{I}] = \left(\frac{2}{N}\right)^{1/2} \{\sin\left(kn\frac{\pi}{N}\right)\} \quad (18)$$

$k,n = 1,\ldots, N\text{-}1$

$$[S_N^{II}] = \left(\frac{2}{N}\right)^{1/2} \left\{\epsilon(k)\sin\left[\frac{\pi(2n-1)k}{2N}\right]\right\} \quad (19)$$

$k,n = 1,\ldots, N$

$$[S_N^{III}] = \left(\frac{2}{N}\right)^{1/2} \left\{\epsilon(n)\sin\left[\frac{\pi n(2k-1)}{2N}\right]\right\} \quad (20)$$

$k,n = 1,\ldots, N$

$$[S_N^{IV}] = \left(\frac{2}{N}\right)^{1/2} \left\{\sin\left[\frac{\pi(2n-1)(2k-1)}{4N}\right]\right\} \quad (21)$$

$k,n = 1,\ldots, N$

$$\text{where } \epsilon(i) = \left[\begin{array}{ll}\frac{1}{\sqrt{2}} & \text{for } i=0 \text{ or } N\\ 1 & \text{otherwise.}\end{array}\right.$$

The superscripts and subscripts represent, respectively, the type and the size of the transform.

LITERATURE SURVEY

Several algorithms are reported in literature for efficient implementation of the DCT and DST in general – purpose computers, and also in dedicated VLSI. In the following, we outline some of the important algorithms and architectures for fast implementation of the DCT and DST in general-purpose computers. When Jain [19] first introduced the DST in 1976, he indicated that an N- point DST could be computed using a 2 N – point fast Fourier transform (FFT). The direct computation of each version of the DSTs requires about N^2 multiplications and N (N-1) additions of real numbers. In 1978 , Narasimha and Peterson [30] introduced an algorithm by rearranging the input sequence so that the DCT of N- point sequence could be obtained through computing an N – point FFT of rearranged sequences, which increased the efficiency of computation of the DCT by more than two times . In 1986, Malvar [28] showed the DCT could be computed through the fast Hartley transform (FHT) of the same length. Because the FHT is a transform of real sequence like DCT, they claimed that the FHT based scheme would save another 50% of computation. This type of fast computation of DCT has been along the line of using other fast transforms to obtain the DCT . These algorithms can also be used for computation of DST. Both direct and indirect approaches have a common feature that they all focus on the butterfly structure and aim at reducing number of multiplications and additions. Yip and Wang [39] proposed a prime-factor DST algorithm which required only real number multiplications but its index mapping was complicated. Lee [25] proposed another index mapping scheme for prime-factor DCT which is more efficient compared with that of Yip and Wang [39] . The input index mapping of Lee [25] , however, would not be feasible in variable size applications and it requires extra memory for constructing and combining two index tables. Chen *et al.* [9] developed a fast algorithm which provides a factor of six improvement in computational complexity when compared to conventional DCT and DST algorithms using the FFT . Makhoul showed the N-point DST can be derived by taking the DFT of a $2N$-point even extension of the signal. Haque [17] developed a two-dimensional fast cosine transform (2-D FCT) algorithm for $2^m \times 2^n$ data point. Hou [18] presented a recursive algorithm for DCT with a structure that shows the generation of the higher order DCT matrices directly from the lower order DCT matrices. Gupta and Rao [16] developed a recursive algorithm for DST which is similar to the recursive algorithm for the DCT developed by Hou [18]. Chan and Ho [4] reviewed some direct methods for computing the DCT. They presented a variant of Hou's algorithm which is both in place and numerically stable. They generalized the method using the concept of decimation and orthogonal properties to compute the entire class of discrete sinusoidal transforms. Chan and Ho [5] presented efficient methods for mapping odd-length type-II and type - III DCTs to a real – valued DFT by an index mapping using permutations and sign changes only, and similar mapping was introduced to convert type -IV DCT to real –valued DFT up to a scaling factor.

Along with the growth of integrated circuit technology, high-performance application- specific dedicated processors are evolving rapidly for digital signal processing applications [20,21,22,23]. The VLSI systems yield high throughput of results by maximizing the processing concurrency, so that they provide less expensive and more suitable alternative to general – purpose computers, for real – time and on-line applications. Systolic architectures are emerging as the most popular and dominant class VLSI structures due to simplicity of their processing elements (PEs), modularity of their structures, regular and nearest neighbor interconnections between the PEs, high level of pipelinability, small chip-area and low power dissipation [21]. In systolic architectures, the desired data are pumped rhythmically in a regular interval across the PEs, for yielding high throughput of result by fully pipelined processing. A systolic system consists of a set of interconnected cells or PEs. Each cell is capable of performing some simple operations. Simple, regular communication and control structures have substantial advantages over complicated ones in design and implementation. Cells in a systolic system are typically interconnected to form a systolic array or systolic tree. Information in a systolic system flows between cells in a pipelined fashion and communication with the outside occurs only at the boundary cells. The basic principle of a systolic architecture is that a single PE is replaced by an array of PEs so that higher computation throughput can be achieved without increasing memory bandwidth. Once data is brought out from the memory, it can be used effectively at each PE it passes while being pumped from PE to PE along the array. This is possible for a wide class of compute-bound computations where multiple operations are performed on each data in a repetitive manner. Synchronous design for pumping of data is usually preferred for reasons of simplicity. Achieving high computation throughput using each input data a number of times with only modest memory bandwidth is one of the many advantages of systolic approach. Several systolic algorithms and architectures are, therefore, suggested in literature for VLSI implementation of digital filters, discrete orthogonal transforms, interpolation, convolution and correlation for signal processing problems. The fast algorithms for general purpose computers are, however, not suitable for VLSI implementation due to global communication requirement. Therefore, various algorithms and architectures have been developed for massively parallel implementation of the DCT and DST in VLSI chips.

Chan and Ho [5, 6] and Cho and Lee [10, 11] suggested implementation of prime-factor DCT based on variant DCT structures which required additional complex number multiplications. Chakrabarti and Ja´Ja´ [3] developed a systolic architecture for implementing Lee's algorithm [25]. They wanted to compute the DCT from the DHT. So they modified the index mappings which are essentially the same as Lee's. However, they did not discuss the actual implementation of these index mappings. Lee and Huang [26] suggested a scheme for prime-factor decomposition of the DCT which involves simpler and more efficient index mapping compared with those of [24, 25, 36, and 37] and is devoid of complex arithmetic operations as well. Also they proposed two systolic architectures comprising of two matrix multiplication units and a transposition unit. Bit-level systolic arrays are more regular, and require simpler PEs (comprised of only gated full adders) compared with word –level PEs, so that higher throughput computation can be achieved by bit-level systolic arrays. Chakrabarti and Ja´Ja´ [3] have suggested a bit level architecture for computation of prime-factor DCT. Mc. Govern *et al.* [29] suggested a bit-serial architecture for implementation of 8 x 8 point 2-D DCT which is not modular and requires complicated interconnections. Chau and Siu [7] proposed an algorithm for implementation of DCT of any general length using a recursive filter structure. They have claimed that the algorithm can be implemented in a regular structure. But, due to the recursive nature of the algorithm, truncation error would accumulate in each stage of recursion so that the transformed output may contain significant amount of error. Chou and Siu [8] showed that by some suitable mappings prime-length DCT may be converted into two suitable transforms with approximately half the original length and the DCT may be converted directly into recursive filter structure that required only constant multiplier for the computation. Gou *et al*. [15] have suggested a systolic architecture for prime-length DCT using input/output data permutation and symmetry property of cosine kernels. But the over heads of this array include some additional shift registers, latches, multiplexers, a demultiplexer and a switching element for control requirement. Sun *et al.* [35] have presented a regular and efficient, I.C. realization for DCT by concurrent architecture using distributed arithmetic and memory oriented structure. The ROM size of this architecture increases rapidly with the order of the DCT so that it may be useful for implementation of the DCT of lower order only. Duhamel and H'Mida [14] have derived a two step algorithm that converts the DCT into a set of circular convolutions. A new algorithm to convert DCT to skew-circular convolution was presented by Li [27] because VLSI implementation of distributed arithmetic is very efficient for computing convolutions. Duh and Wu [13] presented a two stage algorithm and its corresponding architectures for efficient computation of a power-of-two length DCT in which the transform matrix is decomposed into the product of two matrices, the preprocessing and the postprocessing ones.

A detailed study of the available literature reveals that possibly new algorithms may be developed for implementing the DCT and DST in dedicated VLSI more efficiently compared with the existing algorithms.

REFERRENCES

[1] AHMED, N., NATARAJAN, T., and RAO, K.R.: ‘Discrete cosine transforms’, *IEEE Trans. on Computers,* vol. C-23, pp. 90-93, Jan. 1974.

[2] ANDREWS, H.C.: ‘Multi-dimensional rotations in feature selection’, *IEEE Trans. on Computers,* vol . C-20, pp. 1405 -1405, Sept. 1971.

[3] CHAKRABARTI, C. and JA’ JA’, J. : ‘Systolic architectures for the computation of the discrete Hartley and the discrete cosine transforms based on prime-factor decomposition’, *IEEE Trans. on Computers,* vol. 39, No. 11, pp. 1359-1368, Nov. 1990.

[4] CHAN, S. C. and HO, K.L.: ‘Direct methods for computing discrete sinusoidal transforms’, *IEE Processing,* vol. 137, No. 6, pp. 433-442, Dec. 1990.

[5] CHAN, S.C. and HO, K.L.: ‘Fast algorithms for computing the discrete cosine transform’, *IEEE Trans. on Circuits & Systems,* vol. 39, No. 3, pp. 185-190, March 1992.

[6] CHAN, S.C. and HO, K. L.: ‘Prime factor real-valued Fourier, cosine, and Hartley transforms’, *Sixth European Signal Processing Conf.* (Brussels, Belgium), pp. 1045-1048, August 1992.

[7] CHAU, L.P. and SIU, W. C. : ‘Recursive algorithm for the discrete cosine transform with general lengths’, *Electron. Lett.,* vol. 30, No. 3, pp. 197-198, Feb. 1994.

[8] CHAU, L. P. and SIU, W. C. : ‘Direct formulation for the realisation of discrete cosine transform using recursive structure’, *IEEE Trans. on Circuits & Systems*-II : *Analog and Digital Signal Processing,* vol. 42, No. 1, pp. 50-51, Jan. 1995.

[9] CHEN, W.H., SMITH, C.H., and FRALICK, S.C. : ‘A fast computational algorithm for the discrete cosine transform’, *IEEE Trans. on Commun.*, vol. COM-25, No.9, pp. 1004-1009, Sept. 1977.

[10] CHO, N.I. and LEE, S.U.: ‘Fast algorithm and implementation of 2-D discrete cosine transform’, *IEEE Trans. on Circuits & Systems,* vol. 38, pp. 297-305, Mar. 1991.

[11] CHO, N.I. and LEE, S.U.: ‘ A fast 4x4 algorithm for the recursive 2-D DCT’, *IEEE Trans. on Signal Processing,* vol.40, pp.2166-2172, 1992.

[12] CLARKE, R.J.: 'Relation between the Karhunen-Loeve and cosine transforms', *IEE Proc.*, vol. 128, No.6, pp. 359-360, Nov. 1981.

[13] DUH, W.J. and WU, J.L.: 'Two-stage circular-convolution like algorithm/architecture for the discrete cosine transform', *IEE Proceedings*, vol. 137, No.6, pp. 465-472, Dec. 1990.

[14] DUHAMEL, P. and H'MIDA, H.: ' New 2^n DCT algorithms suitable for VLSI implementation', *Proc. ICASSP' 87*, (Dallas), pp. 1805 – 1808, 1987.

[15] GUO, J.I.: LIU, C.M., and JEN, C.W.: 'A new array architecture for prime-length discrete cosine transform', *IEEE Trans. on Signal Processing*, vol. 41, No.1, pp. 436-442, Jan. 1993.

[16] GUPTA, A. and RAO, K.R.: 'A fast recursive algorithm for the discrete sine transform', *IEEE Trans. Acoust., Speech, & Signal Processing*, vol. ASSP – 38, pp. 553-557, 1990.

[17] HAQUE , M.A.: ' A two – dimensional fast cosine transform', *IEEE Trans. on Acoustics, Speech, & Signal Processing*, vol. ASSP-33 , No.6, pp. 1532-1539, Dec. 1985.

[18] HOU,H.S.: 'A fast recursive algorithm for computing the discrete cosine transform', *IEEE Trans. on Acoustics, Speech, & Signal Processing* , vol. ASSP- 35, No.10, pp. 1455-1461, Oct. 1987.

[19] JAIN, A.K. : 'A fast Karhunen-Loeve transform for a class of stochastic process', *IeEE Trans . on Commun.*, vol . COM-24, pp. 1023-1029, 1976.

[20] KUNG, H.T. : ' Why systolic architectures?', *IEEE Trans. on Computers*, vol. 15, No. 1, pp. 37 – 46, January 1982.

[21] KUNG , H.T. and LEISERSON , C.E. : ' *Introduction to VLSI Systems'*, Reading , M.A.: Addision – Wesley, 1980.

[22] KUNG, S.Y.: ' *VLSI Array Processors',* Englewood Cliffs, New Jersey : Prentice – Hall, 1988.

[23] KUNG , S.Y., WHITEHOUSE , H.J., and KAILATH , T. (Eds).: *' VLSI and Modern Signal Processing',* Englewood Cliffs, New Jersey: Prentice – Hall, 1985.

[24] LEE, B.G. : ' A new algorithm to compute the discrete cosine transform', *IEEE Trans. on Acoustics, Speech, & Signal Processing*, vol. ASSP – 32 , No. 6, pp. 1243 – 1245 , December 1984.

[25] LEE , B.G. : ‘ Input and output index mappings for a prime – factor decomposed computation of discrete cosine transform’, *IEEE Trans. on Acoustics, Speech, & Signal Processing,* vol. ASSP- 37, No. 2, pp. 237 -244, Feb. 1989.

[26] LEE, P.Z. and HUANG, F.Y.: ‘ An efficient prime- factor algorithm for the discrete cosine transform and its hardware implementation’, *IEEE Trans. on Signal Processing* , vol. 42, No. 8, pp. 1996 – 2005, August 1994.

[27] Li, W.: ‘ A new algorithm to compute the DCT and its inverse’, *IEEE Trans. on Signal Processing,* vol. 39, No. 6, pp. 1305 - 1313, June 1991.

[28] MALVAR, H.S. : ‘ Fast computation of discrete cosine transform through fast Hartley transform’, *Electron. Lett.*, vol. 22, pp 353 – 354 , March 1986.

[29] MC GOVERN, F.A., WOODS, R.F., and YAN, M.: ‘Novel VLSI implementation of (8 x 8) point 2-D DCT’, *Electron. Lett.*, vol. 30, No. 8, pp. 624 – 626, April 1994.

[30] NARASIMHA, M.J. and PETERSON, A.M.: ‘ On the computation of the discrete cosine transform’, *IEEE Trans. on Commun.,* vol. COM – 26 , pp. 934 – 936 , June 1978.

[31] PEARL, J., ANDREWS, H.C., and PRATT, W.K.: ‘ Performance measures for transform data coding’, *IEEE Trans . on Commun. Technol.,* vol. COM – 20 , pp. 411 – 415 , June 1972.

[32] PRATT , W.K.: ‘Generalised Wiener filtering computation techniques’, *IEEE Trans. on Comput.,* vol. C – 21 , pp . 636 – 641 , July 1972.

[33] PRATT, W.K.: ‘Walsh functions in image processing and two – dimensional filtering’, *in Proc. Symp. Applications of Walsh Functions*, AD- 744650, pp.14-22, 1972.

[34] PROAKIS , J.G. and MANOLAKIS , D.G.: *‘Introduction to Digital Signal Processing’*, New York: Macmillan Publishing Co., 1988.

[35] SUN, M.T., WU, L., and LIOU, M.L.: ‘A concurrent architecture for VLSI implementation of discrete cosine transform’, *IEEE Trans. on Circuits & Systems,* vol. CAS – 34 , No.8, pp.992 – 994, Aug. 1987.

[36] WEL, C.H. and CHEN, C.F.: 'On the computation of the length - 2^m discrete cosine and sine transforms via the permuted difference coefficient', *IEEE Trans. on Signal Processing*, vol. 44, No.2 , pp. 387 -396 , February 1996.

[37] YIP,P. and RAO, K.R.: 'A fast computation algorithm for the discrete sine transform' , *IEEE Trans. Commun.*, vol. COM – 28 , pp.304 -307 , 1980.

[38] YIP,P. and RAO, K.R.: 'On the computation and effectiveness of discrete sine transform', *Computers and Elect. Engng*. vol.7, pp. 45- 55, 1980.

[39] YIP,P. and WANG, F.: 'A prime – factor decomposed algorithm for the discrete sine transform', *Computers and Elect. Engng*., vol. 16, No.1, pp. 43 – 49 , 1990.