Comments: Fast Fourier Transforms[*]

C. Sidney Burrus

1 Other work and Results

This section comes from a note describing results on efficient algorithms to calculate the discrete Fourier transform (DFT) that were collected over years. Perhaps the most interesting is the discovery that the Cooley-Tukey FFT was described by Gauss in 1805 [45]. That gives some indication of the age of research on the topic, and the fact that a 1995 compiled bibliography [95] on efficient algorithms contains over 3400 entries indicates its volume. Three IEEE Press reprint books contain papers on the FFT [69], [21], [22]. An excellent general purpose FFT program has been described in [34], [33] and is used in Matlab and available over the internet.

In addition to this book there are several others [59], [65], [7], [43], [107], [64], [8], [10], [86] that give a good modern theoretical background for the FFT, one book [14] that gives the basic theory plus both FORTRAN and TMS 320 assembly language programs, and other books [56], [89], [16] that contain chapters on advanced FFT topics. A good up-to-date, on-line reference with both theory and programming techniques is in [3]. The history of the FFT is outlined in [23], [45] and excellent survey articles can be found in [30], [25]. The foundation of much of the modern work on efficient algorithms was done by S. Winograd. These results can be found in [116], [117], [118]. An outline and discussion of his theorems can be found in [56] as well as [59], [65], [7], [43].

Efficient FFT algorithms for length-2^M were described by Gauss and discovered in modern times by Cooley and Tukey [24]. These have been highly developed and good examples of FORTRAN programs can be found in [14]. Several new algorithms have been published that require the least known amount of total arithmetic [119], [27], [26], [58], [112], [17]. Of these, the split-radix FFT [27], [26], [111], [98] seems to have the best structure for programming, and an efficient program has been written [91] to implement it. A mixture of decimation-in-time and decimation-in-frequency with very good efficiency is given in [77], [78] and one called the Sine-Cosine FT [17]. Recently a modification to the split-radix algorithm has been described [54] that has a slightly better total arithmetic count. Theoretical bounds on the number of multiplications required for the FFT based on Winograd's theories are given in [43], [44]. Schemes for calculating an in-place, in-order radix-2 FFT are given in [4], [5], [51], [105]. Discussion of various forms of unscramblers is given in [12], [76], [48], [31], [75], [114], [120], [100], [73]. A discussion of the relation of the computer architecture, algorithm and compiler can be found in [63], [61]. A modification to allow lengths of $N = q\,2^m$ for q odd is given in [6].

The "other" FFT is the prime factor algorithm (PFA) which uses an index map originally developed by Thomas and by Good. The theory of the PFA was derived in [55] and further developed and an efficient in-order and in-place program given in [13], [14]. More results on the PFA are given in [103], [104], [105], [106], [96]. A method has been developed to use dynamic programming to design optimal FFT programs

[*] Version 1.8: Sep 18, 2009 2:53 pm -0500

[†] http://creativecommons.org/licenses/by/2.0/

that minimize the number of additions and data transfers as well as multiplications [50]. This new approach designs custom algorithms for a particular computer architecture. An efficient and practical development of Winograd's ideas has given a design method that does not require the rather difficult Chinese remainder theorem [56], [52] for short prime length FFT's. These ideas have been used to design modules of length 11, 13, 17, 19, and 25 [49]. Other methods for designing short DFT's can be found in [102], [57]. A use of these ideas with distributed arithmetic and table look-up rather than multiplication is given in [20]. A program that implements the nested Winograd Fourier transform algorithm (WFTA) is given in [59] but it has not proven as fast or as versatile as the PFA [13]. An interesting use of the PFA was announced [18] in searching for large prime numbers.

These efficient algorithms can not only be used on DFT's but on other transforms with a similar structure. They have been applied to the discrete Hartley transform [92], [9] and the discrete cosine transform [112], [115], [72].

The fast Hartley transform has been proposed as a superior method for real data analysis but that has been shown not to be the case. A well-designed real-data FFT [93] is always as good as or better than a well-designed Hartley transform [92], [29], [68], [109], [101]. The Bruun algorithm [11], [99] also looks promising for real data applications as does the Rader-Brenner algorithm [71], [19], [109]. A novel approach to calculating the inverse DFT is given in [28].

General length algorithms include [84], [36], [32]. For lengths that are not highly composite or prime, the chirp z-transform in a good candidate [14], [70] for longer lengths and an efficient order-N^2 algorithm called the QFT [85], [39], [40] for shorter lengths. A method which automatically generates near-optimal prime length Winograd based programs has been given in [52], [79], [80], [81], [82]. This gives the same efficiency for shorter lengths (i.e. $N \leq 19$) and new algorithms for much longer lengths and with well-structured algorithms. Another approach is given in [66]. Special methods are available for very long lengths [46], [97]. A very interesting general length FFT system called the FFTW has been developed by Frigo and Johnson at MIT. It uses a library of efficient “codelets" which are composed for a very efficient calculation of the DFT on a wide variety of computers [34], [33], [35]. For most lengths and on most computers, this is the fastest FFT today. Surprisingly, it uses a recursive program structure. The FFTW won the 1999 Wilkinson Prize for Numerical Software.

The use of the FFT to calculate discrete convolution was one of its earliest uses. Although the more direct rectangular transform [2] would seem to be more efficient, use of the FFT or PFA is still probably the fastest method on a general purpose computer or DSP chip [67], [93], [29], [60]. On special purpose hardware or special architectures, the use of distributed arithmetic [20] or number theoretic transforms [1] may be even faster. Special algorithms for use with the short-time Fourier transform [87] and for the calculation of a few DFT values [90], [74], [88] and for recursive implementation [113], [33] have also been developed. An excellent analysis of efficient programming the FFT on DSP microprocessors is given in [62], [61]. Formulations of the DFT in terms of tensor or Kronecker products look promising for developing algorithms for parallel and vector computer architectures [94], [107], [53], [110], [108], [38], [37].

Various approaches to calculating approximate DFTs have been based on cordic methods, short word lengths, or some form of pruning. A new method that uses the characteristics of the signals being transformed has combined the discrete wavelet transform (DWT) combined with the DFT to give an approximate FFT with $O(N)$ multiplications [41], [42], [15] for certain signal classes. A similar approach has been developed using filter banks [83], [47].

The study of efficient algorithms not only has a long history and large bibliography, it is still an exciting research field where new results are used in practical applications.

More information can be found on the Rice DSP Group's web page[1]

[1] http://www-dsp.rice.edu

References

[1] R. C. Agarwal and C. S. Burrus. Number theoretic transforms to implement fast digital convolution. *Proceedings of the IEEE*, 63(4):5508211;560, April 1975. also in IEEE Press DSP Reprints II, 1979.

[2] R. C. Agarwal and J. W. Cooley. New algorithms for digital convolution. *IEEE Trans. on ASSP*, 25(2):3928211;410, October 1977.

[3] Jrg Arndt. *Algorithms for Programmers: Ideas and Source Code*. http://www.jjj.de/fxt/, Bayreuth, Germany, 2008. FFT book available on-line, 1000 pages, continually updated.

[4] James K. Beard. An in-place, self-reordering fft. In *Proceedings of the ICASSP-78*, pages 632–633, Tulsa, April 1978.

[5] James K. Beard. *The FFT in the 21st Century: Eigenspace Processing*. Kluwer, Boston, 2003.

[6] Guoan Bi and Yan Qiu Chen. Fast dft algorithms for length. *IEEE Transactions on Circuits and Systems 8211; II*, 45(6):6858211;690, June 1998.

[7] R. E. Blahut. *Fast Algorithms for Digital Signal Processing*. Addison-Wesley, Inc., Reading, MA, 1984.

[8] Richard E. Blahut. *Algebraic Methods for Signal Processing and Communications Coding*. Springer-Verlag, New York, 1992.

[9] R. N. Bracewell. *The Hartley Transform*. Oxford Press, 1986.

[10] William L. Briggs and Van Emden Henson. *The DFT: An Owner's Manual for the Discrete Fourier Transform*. SIAM, Philadelphia, 1995.

[11] G. Bruun. z-transform dft filters and ffts. *IEEE Transactions on ASSP*, 26(1):568211;63, February 1978.

[12] C. S. Burrus. Unscrambling for fast dft algorithms. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 36(7):10868211;1087, July 1988.

[13] C. S. Burrus and P. W. Eschenbacher. An in8211;place, in8211;order prime factor fft algorithm. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 29(4):8068211;817, August 1981. Reprinted in it DSP Software, by L.R. Morris, 1983.

[14] C. S. Burrus and T. W. Parks. *DFT/FFT and Convolution Algorithms*. John Wiley & Sons, New York, 1985.

[15] C. Sidney Burrus, Ramesh A. Gopinath, and Haitao Guo. *Introduction to Wavelets and the Wavelet Transform*. Prentice Hall, Upper Saddle River, NJ, 1998.

[16] C. Sidney Burrus and Ivan W. Selesnick. Fast convolution and filtering. In V. K. Madisetti and D. B. Williams, editors, *The Digital Signal Processing Handbook*, chapter 8. CRC Press, Boca Raton, 1998.

[17] Iain R. Byam. A new fast fourier transform algorithm. Technical report, University of the West Indies (UWI), St. Augustine, Trinidad, June 1999. A short version is in Technical Report PG-99001, ECE Dept., Univ. of the West Indies, Aug. 1999.

[18] K. T. Chen. A new record: The largest known prime number. *IEEE Spectrum*, 27(2):47, February 1990.

[19] K. M. Cho and G. C. Temes. Real-factor fft algorithms. In *Proceedings of IEEE ICASSP-78*, pages 634–637, Tulsa, OK, April 1978.

[20] Shuni Chu and C. S. Burrus. A prime factor fft algorithm using distributed arithmetic. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 30(2):2178211;227, April 1982.

[21] DSP Committee, editor. *Digital Signal Processing II, selected reprints.* IEEE Press, New York, 1979.

[22] DSP Committee, editor. *Programs for Digital Signal Processing.* IEEE Press, New York, 1979.

[23] J. W. Cooley. How the fft gained acceptance. *IEEE Signal Processing Magazine*, 9(1):108211;13, January 1992. Also presented at the ACM Conference on the History of Scientific and Numerical Computation, Princeton, NJ, May 1987 and published in: A History of Scientific Computing, edited by S. G. Nash, Addison-Wesley, 1990, pp. 133-140.

[24] J. W. Cooley and J. W. Tukey. An algorithm for the machine calculation of complex fourier series. *Math. Computat.*, 19:2978211;301, 1965.

[25] James W. Cooley. The structure of fft algorithms, April 1990. Notes for a Tutorial at IEEE ICASSP-90.

[26] P. Duhamel. Implementation of 'split-radix' fft algorithms for complex, real, and real-symmetric data. *IEEE Trans. on ASSP*, 34:2858211;295, April 1986. A shorter version appeared in the ICASSP-85 Proceedings, p.20.6, March 1985.

[27] P. Duhamel and H. Hollmann. Split radix fft algorithm. *Electronic Letters*, 20(1):148211;16, January 5 1984.

[28] P. Duhamel, B. Piron, and J. M. Etcheto. On computing the inverse dft. *IEEE Transactions on ASSP*, 36(2):2858211;286, February 1978.

[29] P. Duhamel and M. Vetterli. Improved fourier and hartley transfrom algorithms, application to cyclic convolution of real data. *IEEE Trans. on ASSP*, 35(6):8188211;824, June 1987.

[30] P. Duhamel and M. Vetterli. Fast fourier transforms: A tutorial review and a state of the art. *Signal Processing*, 19(4):2598211;299, April 1990.

[31] D. M. W. Evans. A second improved digit8211;reversal permutation algorithm for fast transforms. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 37(8):12888211;1291, August 1989.

[32] W. E. Ferguson, Jr. A simple derivation of glassman general-n fast fourier transform. *Comput. and Math. with Appls.*, 8(6):4018211;411, 1982. Also, in Report AD-A083811, NTIS, Dec. 1979.

[33] Matteo Frigo. A fast fourier transform compiler. In *Proceedings of the 1999 ACM SIGPLAN Conference on Programming Language Design and Implentation*, PLDI-99, Atlanta, May 1999.

[34] Matteo Frigo and Steven G. Johnson. Fftw: An adaptive software architecture for the fft. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, volume III, page 13818211;1384, ICASSP-98, Seattle, May 128211;15 1998.

[35] Matteo Frigo and Steven G. Johnson. The design and implementtion of fftw. *Proceedings of the IEEE*, 93(2):2168211;231, February 2005.

[36] J. A. Glassman. A generalization of the fast fourier transform. *IEEE Transactions on Computers*, C-19(2):1058211;116, Feburary 1970.

[37] John Granata, Michael Conner, and Richard Tolimieri. Recursive fast algorithms and the role of the tensor product. *IEEE Transactions on Signal Processing*, 40(12):29218211;2930, December 1992.

[38] John Granata, Michael Conner, and Richard Tolimieri. The tensor product: A mathematical programming language for ffts. *IEEE Signal Processing Magazine*, 9(1):408211;48, January 1992.

[39] H. Guo, G. A. Sitton, and C. S. Burrus. The quick discrete fourier transform. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, page III:4458211;448, IEEE ICASSP-94, Adelaide, Australia, April 198211;22 1994.

[40] H. Guo, G. A. Sitton, and C. S. Burrus. The quick fourier transform: an fft based on symmetries. *IEEE Transactions on Signal Processing*, 46(2):3358211;341, February 1998.

[41] Haitao Guo and C. Sidney Burrus. Approximate fft via the discrete wavelet transform. In *Proceedings of SPIE Conference 2825*, Denver, August 68211;9 1996.

[42] Haitao Guo and C. Sidney Burrus. Wavelet transform based fast approximate fourier transform. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, volume 3, page III:19738211;1976, IEEE ICASSP-97, Munich, April 218211;24 1997.

[43] M. T. Heideman. *Multiplicative Complexity, Convolution, and the DFT*. Springer8211;Verlag, 1988.

[44] M. T. Heideman and C. S. Burrus. On the number of multiplications necessary to compute a length-dft. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 34(1):918211;95, February 1986.

[45] M. T. Heideman, D. H. Johnson, and C. S. Burrus. Gauss and the history of the fft. *IEEE Acoustics, Speech, and Signal Processing Magazine*, 1(4):148211;21, October 1984. also in Archive for History of Exact Sciences, 1985.

[46] W. K. Hocking. Performing fourier transforms on extremely long data streams. *Computers in Physics*, 3(1):598211;65, January 1989.

[47] A. N. Hossen, U. Heute, O. V. Shentov, and S. K. Mitra. Subband dft 8211; part ii: Accuracy, complexity, and applications. *Signal Processing*, 41:2798211;295, 1995.

[48] Jechang Jeong and William J. Williams. A fast recursive bit-reversal algorithm. In *Proceedings of the ICASSP-90*, page 15118211;1514, Albuquerque, NM, April 1990.

[49] H. W. Johnson and C. S. Burrus. Large dft modules: N = 11, 13, 17, 19, and 25. Technical report 8105, Department of Electrical Engineering, Rice University, Houston, TX 772518211;1892, 1981.

[50] H. W. Johnson and C. S. Burrus. The design of optimal dft algorithms using dynamic programming. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 31(2):3788211;387, April 1983.

[51] H. W. Johnson and C. S. Burrus. An in-place, in-order radix-2 fft. In *ICASSP-84 Proceedings*, page 28A.2, March 1984.

[52] Howard W. Johnson and C. S. Burrus. On the structure of efficient dft algorithms. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 33(1):2488211;254, February 1985.

[53] J. Johnson, R. W. Johnson, D. Rodriguez, and R. Tolimieri. A methodology for designing, modifying, and implementing fourier transform algorithms on various architectures. *Circuits, Systems and Signal Processing*, 9(4):4498211;500, 1990.

[54] Steven G. Johnson and Matteo Frigo. A modified split-radix fft with fewer arithmetic operations. *IEEE Transactions on Signal Processing*, 55(1):1118211;119, January 2007.

[55] D. P. Kolba and T. W. Parks. A prime factor fft algorithm using high speed convolution. *IEEE Trans. on ASSP*, 25:2818211;294, August 1977. also in.

[56] Jae S. Lim and A. V. Oppenheim. *Advanced Topics in Signal Processing*, chapter 4. Prentice-Hall, 1988.

[57] Chao Lu, James W. Cooley, and Richard Tolimieri. Fft algorithms for prime transform sizes and their implementations on vax, ibm3090vf, and ibm rs/6000. *IEEE Transactions on Signal Processing*, 41(2):6388211;648, February 1993.

[58] J. B. Martens. Recursive cyclotomic factorization 8211; a new algorithm for calculating the discrete fourier transform. *IEEE Trans. on ASSP*, 32(4):7508211;762, August 1984.

[59] J. H. McClellan and C. M. Rader. *Number Theory in Digital Signal Processing.* Prentice-Hall, Inc., Englewood Cliffs, NJ, 1979.

[60] R. Meyer, R. Reng, and K. Schwarz. Convolution algorithms on dsp processors. In *Proceedings of the ICASSP-91*, page 21938211;2196, Toronto, Canada, May 1991.

[61] R. Meyer and K. Schwarz. Fft implementation on dsp-chips, Sept. 18 1990. preprint.

[62] R. Meyer, K. Schwarz, and H. W. Schuessler. Fft implementation on dsp-chips 8212; theory and practice. In *Proceedings of the ICASSP-90*, page 15038211;1506, Albuquerque, NM, April 1990.

[63] L. R. Morris. *Digital Signal Processing Software.* DSPSW, Inc., Toronto, Canada, 1982, 1983.

[64] Douglas G. Myers. *Digital Signal Processing, Efficient Convolution and Fourier Transform Techniques.* Prentice-Hall, Sydney, Australia, 1990.

[65] H. J. Nussbaumer. *Fast Fourier Transform and Convolution Algorithms.* Springer-Verlag, Heidelberg, Germany, second edition, 1981, 1982.

[66] F. Perez and T. Takaoka. A prime factor fft algorithm implementation using a program generation technique. *IEEE Transactions on Acoustics, Speech and Signal Processing*, 35:12218211;1223, August 1987.

[67] I. Pitas and C. S. Burrus. Time and error analysis of digital convolution by rectangular transforms. *Signal Processing*, 5(2):1538211;162, March 1983.

[68] Miodrag Popovi263; and Dragutin 352;evi263;. A new look at the comparison of the fast hartley and fourier transforms. *IEEE Transactions on Signal Processing*, 42(8):21788211;2182, August 1994.

[69] L. R. Rabiner and C. M. Rader, editors. *Digital Signal Processing, selected reprints.* IEEE Press, New York, 1972.

[70] L.R. Rabiner, R.W. Schafer, and C.M. Rader. The chirp z-transform algorithm. *IEEE Transactions on Audio Electroacoustics*, AU-17:868211;92, June 1969.

[71] C. M. Rader and N. M. Brenner. A new principle for fast fourier transformation. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, ASSP-24(3):264–266, June 1976.

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

[73] J. M. Rius and R. De Porrata-D[U+FFFD]. New fft bit-reversal algorithm. *IEEE Transactions on Signal Processing*, 43(4):9918211;994, April 1995.

[74] Christian Roche. A split8211;radix partial input/output fast fourier transform algorithm. *IEEE Transactions on Signal Processing*, 40(5):12738211;1276, May 1992.

[75] J. J. Rodr[U+FFFD]ez. An improved fft digit8211;reversal algorithm. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 37(8):12988211;1300, August 1989.

[76] Petr Rsel. Timing of some bit reversal algorithms. *Signal Processing*, 18(4):4258211;433, December 1989.

[77] Ali Saidi. Decimation-in-time-frequency fft algorithm. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, volume 3, page III:4538211;456, IEEE ICASSP-94, Adelaide, Australia, April 198211;22 1994.

[78] Ali Saidi. Decimation-in-time-frequency fft algorithm, 1996. manuscript.

[79] I. W. Selesnick and C. S. Burrus. Automating the design of prime length fft programs. In *Proceedings of the IEEE International Symposium on Circuits and Systems*, volume 1, page 1338211;136, ISCAS-92, San Diego, CA, May 1992.

[80] I. W. Selesnick and C. S. Burrus. Multidimensional mapping techniques for convolution. In *Proceedings of the IEEE International Conference on Signal Processing*, volume III, pages III–2888211;291, IEEE ICASSP-93, Minneapolis, April 1993.

[81] I. W. Selesnick and C. S. Burrus. Extending winograd's small convolution algorithm to longer lengths. In *Proceedings of the IEEE International Symposium on Circuits and Systems*, volume 2, page 2.4498211;452, IEEE ISCAS-94, London, June 30 1994.

[82] Ivan W. Selesnick and C. Sidney Burrus. Automatic generation of prime length fft programs. *IEEE Transactions on Signal Processing*, 44(1):148211;24, January 1996.

[83] O. V. Shentov, S. K. Mitra, U. Heute, and A. N. Hossen. Subband dft 8211; part i: Definition, interpretation and extensions. *Signal Processing*, 41:2618211;278, 1995.

[84] R.C. Singleton. An algorithm for computing the mixed radix fast fourier transform. *IEEE Transactions on Audio and Electroacoustics*, AU-17:93–103, June 1969.

[85] G. A. Sitton. The qft algorithm, 1985.

[86] Winthrop W. Smith and Joanne M. Smith. *Handbook of Real-Time Fast Fourier Transforms*. IEEE Press, New York, 1995.

[87] H. V. Sorensen and C. S. Burrus. Efficient computation of the short-time fft. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, pages 1894–1897, New York, April 1988.

[88] H. V. Sorensen and C. S. Burrus. Efficient computation of the dft with only a subset of input or output points. *IEEE Transactions on Signal Processing*, 41(3):11848211;1200, March 1993.

[89] H. V. Sorensen and C. S. Burrus. Fast dft and convolution algorithms. In Sanjit K. Mitra and James F. Kaiser, editors, *Handbook for Digital Signal Processing*, chapter 8. John Wiley & Sons, New York, 1993.

[90] H. V. Sorensen, C. S. Burrus, and D. L. Jones. A new efficient algorithm for computing a few dft points. In *Proceedings of the IEEE International Symposium on Circuits and Systems*, page 19158211;1918, Espoo, Finland, June 1988.

[91] H. V. Sorensen, M. T. Heideman, and C. S. Burrus. On computing the split8211;radix fft. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 34(1):1528211;156, February 1986.

[92] H. V. Sorensen, D. L. Jones, C. S. Burrus, and M. T. Heideman. On computing the discrete hartley transform. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 33(5):12318211;1238, October 1985.

[93] H. V. Sorensen, D. L. Jones, M. T. Heideman, and C. S. Burrus. Real valued fast fourier transform algorithms. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 35(6):8498211;863, June 1987.

[94] H. V. Sorensen, C. A. Katz, and C. S. Burrus. Efficient fft algorithms for dsp processors using tensor product decomposition. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, page 15078211;1510, Albuquerque, NM, April 1990.

[95] Henrik V. Sorensen, C. Sidney Burrus, and Michael T. Heideman. *Fast Fourier Transform Database*. PWS Publishing, Boston, 1995.

[96] R. Stasi324;ski. Prime factor dft algorithms for new small-n dft modules. *IEE Proceedings, Part G*, 134(3):1178211;126, 1987.

[97] R. Stasi324;ski. Extending sizes of effective convolution algorithms. *Electronics Letters*, 26(19):16028211;1604, 1990.

[98] Ryszard Stasi324;ski. The techniques of the generalized fast fourier transform algorithm. *IEEE Transactions on Signal Processing*, 39(5):10588211;1069, May 1991.

[99] R. Storn. On the bruun algorithm and its inverse. *Frequenz*, 46:1108211;116, 1992. a German journal.

[100] D. Sundararajan, M. O. Ahamad, and M. N. S. Swamy. A fast fft bit-reversal algorithm. *IEEE Transactions on Circuits and Systems, II*, 41(10):7018211;703, October 1994.

[101] D. Sundararajan, M. O. Ahmad, and M. N. S. Swamy. Fast computations of the discrete fourier transform of real data. *IEEE Transactions on Signal Processing*, 45(8):20108211;2022, August 1997.

[102] Clive Temperton. A new set of minimum-add small-n rotated dft modules. *Journal of Computational Physics*, 75:1908211;198, 1988.

[103] Clive Temperton. A self-sorting in-place prime factor real/half-complex fft algorithm. *Journal of Computational Physics*, 75:1998211;216, 1988.

[104] Clive Temperton. Nesting strategies for prime factor fft algorithms. *Journal of Computational Physics*, 82(2):2478211;268, June 1989.

[105] Clive Temperton. Self-sorting in-place fast fourier transforms. *SIAM Journal of Sci. Statist. Comput.*, 12(4):8088211;823, 1991.

[106] Clive Temperton. A generalized prime factor fft algorithm for any. *SIAM Journal of Scientific and Statistical Computing*, 13:6768211;686, May 1992.

[107] Richard Tolimieri, Myoung An, and Chao Lu. *Algorithms for Discrete Fourier Transform and Convolution*. Springer-Verlag, New York, second edition, 1989, 1997.

[108] Richard Tolimieri, Myoung An, and Chao Lu. *Mathematics of Multidimensional Fourier Transform Algorithms*. Springer-Verlag, New York, second edition, 1993, 1997.

[109] P. R. Uniyal. Transforming real-valued sequences: Fast fourier versus fast hartley transform algorithms. *IEEE Transactions on Signal Processing*, 42(11):32498211;3254, November 1994.

[110] Charles Van Loan. *Matrix Frameworks for the Fast Fourier Transform*. SIAM, Philadelphia, PA, 1992.

[111] Martin Vetterli and P. Duhamel. Split-radix algorithms for length- dft's. *IEEE Trans. on ASSP*, 37(1):578211;64, January 1989. Also, ICASSP-88 Proceedings, pp.14158211;1418, April 1988.

[112] Martin Vetterli and H. J. Nussbaumer. Simple fft and dct algorithms with reduced number of operations. *Signal Processing*, 6(4):2678211;278, August 1984.

[113] A. R. V[U+FFFD]onyi-K[U+FFFD]. A recursive fast fourier transformation algorithm. *IEEE Transactions on Circuits and Systems, II*, 42(9):6148211;616, September 1995.

[114] J. S. Walker. A new bit reversal algorithm. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 38(8):14728211;1473, August 1990.

[115] Fang Ming Wang and P. Yip. Fast prime factor decomposition algorithms for a family of discrete trigonometric transforms. *Circuits, Systems, and Signal Processing*, 8(4):4018211;419, 1989.

[116] S. Winograd. On computing the discrete fourier transform. *Mathematics of Computation*, 32:1758211;199, January 1978.

[117] S. Winograd. On the multiplicative complexity of the discrete fourier transform. *Advances in Mathematics*, 32(2):838211;117, May 1979. also in.

[118] S. Winograd. *Arithmetic Complexity of Computation*. SIAM CBMS-NSF Series, No.33. SIAM, Philadelphia, 1980.

[119] R. Yavne. An economical method for calculating the discrete fourier transform. In *Fall Joint Computer Conference, AFIPS Conference Proceedings*, volume 33 part 1, page 1158211;125, 1968.

[120] Angelo A. Yong. A better fft bit-reversal algorithm without tables. *IEEE Transactions on Signal Processing*, 39(10):23658211;2367, October 1991.