

Notes on the FFT

C. S. Burrus

Department of Electrical and Computer Engineering
Rice University, Houston, TX 77251-1892, e-mail: csb@rice.edu

September 29, 1997

This is a note describing results on efficient algorithms to calculate the discrete Fourier transform (DFT). The purpose is to report work done at Rice University, but other contributions used by the DSP research group at Rice are also cited. Perhaps the most interesting is the discovery that the Cooley-Tukey FFT was described by Gauss in 1805 [1]. That gives some indication of the age of research on the topic, and the fact that a recently compiled bibliography [2] on efficient algorithms contains over 3400 entries indicates its volume. Three IEEE Press reprint books contain papers on the FFT [3, 4, 5]. An excellent general purpose FFT program has been described in [6, 7] and is available over the internet.

There are several books [8, 9, 10, 11, 12, 13, 14, 15, 16] that give a good modern theoretical background for this work, one book [17] that gives the basic theory plus both FORTRAN and TMS 320 assembly language programs, and other books [18, 19, 20] that contains a chapter on advanced FFT topics. The history of the FFT is outlined in [21, 1] and excellent survey articles can be found in [22, 23]. The foundation of much of the modern work on efficient algorithms was done by S. Winograd. This can be found in [24, 25, 26]. An outline and discussion of his theorems can be found in [18] as well as [8, 9, 10, 11].

Efficient FFT algorithms for length- 2^M were described by Gauss and discovered in modern times by Cooley and Tukey [27]. These have been highly developed and good examples of FORTRAN programs can be found in [17]. Several new algorithms have been published that require the least known amount of total arithmetic [28, 29, 30, 31, 32]. Of these, the split-radix FFT [29, 30, 33, 34] seems to have the best structure for programming, and an efficient program has been written [35] to implement it. A mixture of decimation-in-time and decimation-in-frequency with very good efficiency is given in [36]. Theoretical bounds on the number of multiplications required for the FFT based on Winograd's theories are given in [11, 37]. Schemes for calculating an in-place, in-order radix-2 FFT are given in [38, 39, 40]. Discussion of various forms of unscramblers is given in [41, 42, 43, 44, 45, 46, 47, 48, 49]. A discussion of the relation of the computer architecture, algorithm and compiler can be found in [50, 51].

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 [52] and further developed and an efficient in-order and in-place program given in [53, 17]. More results on the PFA are given in [54, 55, 40, 56, 57]. A method has been developed to use dynamic programming to design optimal FFT programs that minimize the number of additions and data transfers as well as multiplications [58]. 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 [18, 59] for short prime length FFT's. These ideas have been used to design modules of length 11, 13, 17, 19, and 25 [60]. Other methods for designing short DFT's can be found in [61, 62]. A use of these ideas with distributed arithmetic and table look-up rather than multiplication is given in [63]. A program that implements the nested Winograd Fourier transform algorithm (WFTA) is given in [8] but it has not proven as fast or as versatile as the PFA [53]. An interesting use of the PFA was announced [64] 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 [65, 66] and the discrete cosine transform [32, 67, 68].

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 [69] is always as good as or better than a well-designed Hartley transform [65, 70, 71, 72, 73]. The Bruun algorithm [74, 75] also looks promising for real data applications as does the Rader-Brenner algorithm [76, 77, 72]. A novel approach to calculating the inverse DFT is given in [78].

General length algorithms include [79, 80, 81]. For lengths that are not highly composite or prime, the chirp z-transform is a good candidate [17, 82] for longer lengths and an efficient order- N^2 algorithm called the QFT [83, 84, 85] for shorter lengths. A method which automatically generates near-optimal prime length Winograd based programs has been given in [59, 86, 87, 88, 89]. 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. Special methods are available for very long lengths [90, 91]. A very interesting general length FFT system called the FFTW has been developed by Frigo and Johnson at MIT which uses a library of efficient “codelets” which are composed for a very efficient calculation of the DFT on a wide variety of computers [6, 7]. For most lengths and on most computers, this is the fastest FFT today.

The use of the FFT to calculate discrete convolution was one of its earliest uses. Although the more direct rectangular transform [92] 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 [93, 69, 70, 94] although the use of distributed arithmetic [63] or number theoretic transforms [95] with special hardware may be even faster. Special algorithms for use with the short-time Fourier transform [96] and for the calculation of a few DFT values [97, 98, 99] and for recursive implementation [100] have also been developed. An excellent analysis of efficient programming the FFT on DSP microprocessors is given in [101, 51]. Formulations of the DFT in terms of tensor or Kronecker products look promising for developing algorithms for parallel and vector computer architectures [102, 12, 103, 104, 105, 106, 107].

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 [108, 109, 110, 111] for certain signal classes.

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: <http://www-dsp.rice.edu> and this document can be found at: <http://www-dsp.rice.edu/res/fft/fftnote.asc>.

References

- [1] M. T. Heideman, D. H. Johnson, and C. S. Burrus, “Gauss and the history of the FFT,” *IEEE Acoustics, Speech, and Signal Processing Magazine*, vol. 1, pp. 14–21, October 1984. also in *Archive for History of Exact Sciences*, 1985.
- [2] H. V. Sorensen, C. S. Burrus, and M. T. Heideman, *Fast Fourier Transform Database*. Boston: PWS Publishing, 1995.
- [3] L. R. Rabiner and C. M. Rader, eds., *Digital Signal Processing, selected reprints*. New York: IEEE Press, 1972.
- [4] DSP Committee, ed., *Digital Signal Processing II, selected reprints*. New York: IEEE Press, 1979.
- [5] DSP Committee, ed., *Programs for Digital Signal Processing*. New York: IEEE Press, 1979.
- [6] M. Frigo and S. G. Johnson, “The fastest Fourier transform in the west,” Tech. Rep. MIT-LCS-TR-728, Laboratory for Computer Science, MIT, Cambridge, MA, September 1997.
- [7] M. Frigo and S. G. Johnson, “The fastest Fourier transform in the west,” Laboratory for Computer Science, MIT, 1997. A set of 34 slides obtained over the internet, Sept. 1997.
- [8] J. H. McClellan and C. M. Rader, *Number Theory in Digital Signal Processing*. Englewood Cliffs, NJ: Prentice-Hall, Inc., 1979.
- [9] H. J. Nussbaumer, *Fast Fourier Transform and Convolution Algorithms*. Heidelberg, Germany: Springer-Verlag, second ed., 1981, 1982.
- [10] R. E. Blahut, *Fast Algorithms for Digital Signal Processing*. Reading, MA: Addison-Wesley, Inc., 1984.
- [11] M. T. Heideman, *Multiplicative Complexity, Convolution, and the DFT*. Springer-Verlag, 1988.
- [12] R. Tolimieri, M. An, and C. Lu, *Algorithms for Discrete Fourier Transform and Convolution*. New York: Springer-Verlag, 1989.
- [13] D. G. Myers, *Digital Signal Processing, Efficient Convolution and Fourier Transform Techniques*. Sydney, Australia: Prentice-Hall, 1990.
- [14] R. E. Blahut, *Algebraic Methods for Signal Processing and Communications Coding*. New York: Springer-Verlag, 1992.
- [15] W. L. Briggs and V. E. Henson, *The DFT: An Owner’s Manual for the Discrete Fourier Transform*. Philadelphia: SIAM, 1995.

- [16] W. W. Smith and J. M. Smith, *Handbook of Real-Time Fast Fourier Transforms*. New York: IEEE Press, 1995.
- [17] C. S. Burrus and T. W. Parks, *DFT/FFT and Convolution Algorithms*. New York: John Wiley & Sons, 1985.
- [18] J. S. Lim and A. V. Oppenheim, *Advanced Topics in Signal Processing*, ch. 4. Prentice-Hall, 1988.
- [19] H. V. Sorensen and C. S. Burrus, "Fast DFT and convolution algorithms," in *Handbook for Digital Signal Processing*, (S. K. Mitra and J. F. Kaiser, eds.), ch. 8, New York: John Wiley & Sons, 1993.
- [20] C. S. Burrus and I. W. Selesnick, "Fast convolution and filtering," in *The Digital Signal Processing Handbook*, (V. K. Madiseti and D. B. Williams, eds.), Boca Raton: CRC Press, to appear 1997.
- [21] J. W. Cooley, "How the FFT gained acceptance," *IEEE Signal Processing Magazine*, vol. 9, pp. 10–13, 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.
- [22] P. Duhamel and M. Vetterli, "Fast Fourier transforms: a tutorial review and a state of the art," *Signal Processing*, vol. 19, pp. 259–299, April 1990.
- [23] J. W. Cooley, "The structure of FFT algorithms," April 1990. Notes for a Tutorial at IEEE ICASSP-90.
- [24] S. Winograd, "On computing the discrete Fourier transform," *Mathematics of Computation*, vol. 32, pp. 175–199, January 1978.
- [25] S. Winograd, "On the multiplicative complexity of the discrete Fourier transform," *Advances in Mathematics*, vol. 32, pp. 83–117, May 1979. also in [8].
- [26] S. Winograd, *Arithmetic Complexity of Computation. SIAM CBMS-NSF Series, No. 33*, Philadelphia: SIAM, 1980.
- [27] J. W. Cooley and J. W. Tukey, "An algorithm for the machine calculation of complex Fourier series," *Math. Computat.*, vol. 19, pp. 297–301, 1965.
- [28] R. Yavne, "An economical method for calculating the discrete Fourier transform," in *Proceedings of the Fall Joint Computer Conference*, pp. 115–125, 1968.
- [29] P. Duhamel and H. Hollmann, "Split radix FFT algorithm," *Electronic Letters*, vol. 20, pp. 14–16, January 5 1984.
- [30] P. Duhamel, "Implementation of 'split-radix' FFT algorithms for complex, real, and real-symmetric data," *IEEE Trans. on ASSP*, vol. 34, pp. 285–295, April 1986. A shorter version appeared in the *ICASSP-85 Proceedings*, p. 20.6, March 1985.
- [31] J. B. Martens, "Recursive cyclotomic factorization – a new algorithm for calculating the discrete Fourier transform," *IEEE Trans. on ASSP*, vol. 32, pp. 750–762, August 1984.
- [32] M. Vetterli and H. J. Nussbaumer, "Simple FFT and DCT algorithms with reduced number of operations," *Signal Processing*, vol. 6, pp. 267–278, August 1984.
- [33] M. Vetterli and P. Duhamel, "Split-radix algorithms for length- p^m DFT's," *IEEE Trans. on ASSP*, vol. 37, pp. 57–64, January 1989. Also, *ICASSP-88 Proceedings*, pp. 1415–1418, April 1988.
- [34] R. Stasiński, "The techniques of the generalized fast Fourier transform algorithm," *IEEE Transactions on Signal Processing*, vol. 39, pp. 1058–1069, May 1991.
- [35] H. V. Sorensen, M. T. Heideman, and C. S. Burrus, "On computing the split-radix FFT," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 34, pp. 152–156, February 1986.
- [36] A. Saidi, "Decimation-in-time-frequency FFT algorithm," in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, (IEEE ICASSP-94, Adelaide, Australia), pp. III:453–456, April 19–22 1994.
- [37] M. T. Heideman and C. S. Burrus, "On the number of multiplications necessary to compute a length- 2^n DFT," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 34, pp. 91–95, February 1986.
- [38] J. K. Beard, "An in-place, self-reordering FFT," in *Proceedings of the ICASSP-78*, (Tulsa), pp. 632–633, April 1978.

- [39] H. W. Johnson and C. S. Burrus, "An in-place, in-order radix-2 FFT," in *ICASSP-84 Proceedings*, p. 28A.2, March 1984.
- [40] C. Temperton, "Self-sorting in-place fast Fourier transforms," *SIAM Journal of Sci. Statist. Comput.*, vol. 12, no. 4, pp. 808–823, 1991.
- [41] C. S. Burrus, "Unscrambling for fast DFT algorithms," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 36, pp. 1086–1087, July 1988.
- [42] P. Rösler, "Timing of some bit reversal algorithms," *Signal Processing*, vol. 18, pp. 425–433, December 1989.
- [43] J. Jeong and W. J. Williams, "A fast recursive bit-reversal algorithm," in *Proceedings of the ICASSP-90*, (Albuquerque, NM), pp. 1511–1514, April 1990.
- [44] D. M. W. Evans, "A second improved digit-reversal permutation algorithm for fast transforms," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 37, pp. 1288–1291, August 1989.
- [45] J. J. Rodríguez, "An improved FFT digit-reversal algorithm," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 37, pp. 1298–1300, August 1989.
- [46] J. S. Walker, "A new bit reversal algorithm," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 38, pp. 1472–1473, August 1990.
- [47] A. A. Yong, "A better FFT bit-reversal algorithm without tables," *IEEE Transactions on Signal Processing*, vol. 39, pp. 2365–2367, October 1991.
- [48] D. Sundararajan, M. O. Ahamad, and M. N. S. Swamy, "A fast FFT bit-reversal algorithm," *IEEE Transactions on Circuits and Systems, II*, vol. 41, pp. 701–703, October 1994.
- [49] J. M. Rius and R. De Porrata-Dòria, "New FFT bit-reversal algorithm," *IEEE Transactions on Signal Processing*, vol. 43, pp. 991–994, April 1995.
- [50] L. R. Morris, *Digital Signal Processing Software*. Toronto, Canada: DSPSW, Inc., 1982, 1983.
- [51] R. Meyer and K. Schwarz, "FFT implementation on DSP-chips," Sept. 18 1990. preprint.
- [52] D. P. Kolba and T. W. Parks, "A prime factor FFT algorithm using high speed convolution," *IEEE Trans. on ASSP*, vol. 25, pp. 281–294, August 1977. also in [8].
- [53] C. S. Burrus and P. W. Eschenbacher, "An in-place, in-order prime factor FFT algorithm," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 29, pp. 806–817, August 1981. Reprinted in it DSP Software, by L.R. Morris, 1983.
- [54] C. Temperton, "A self-sorting in-place prime factor real/half-complex FFT algorithm," *Journal of Computational Physics*, vol. 75, pp. 199–216, 1988.
- [55] C. Temperton, "Nesting strategies for prime factor FFT algorithms," *Journal of Computational Physics*, vol. 82, pp. 247–268, June 1989.
- [56] C. Temperton, "A generalized prime factor FFT algorithm for any $n = 2^p 3^q 5^r$," *SIAM Journal of Sci. Stat. Comp.*, 1992. to appear.
- [57] R. Stasiński, "Prime factor DFT algorithms for new small-N DFT modules," *IEE Proceedings, Part G*, vol. 134, no. 3, pp. 117–126, 1987.
- [58] H. W. Johnson and C. S. Burrus, "The design of optimal DFT algorithms using dynamic programming," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 31, pp. 378–387, April 1983.
- [59] H. W. Johnson and C. S. Burrus, "On the structure of efficient DFT algorithms," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 33, pp. 248–254, February 1985.
- [60] H. W. Johnson and C. S. Burrus, "Large DFT modules: $N = 11, 13, 17, 19,$ and 25 ," Tech. Rep. 8105, Department of Electrical Engineering, Rice University, Houston, TX 77251–1892, 1981.
- [61] C. Temperton, "A new set of minimum-add small-n rotated DFT modules," *Journal of Computational Physics*, vol. 75, pp. 190–198, 1988.
- [62] C. Lu, J. W. Cooley, and R. Tolimieri, "FFT algorithms for prime transform sizes and their implementations on VAX, IBM3090VF, and IBM RS/6000," *IEEE Transactions on Signal Processing*, vol. 41, pp. 638–648, February 1993.

- [63] S. Chu and C. S. Burrus, "A prime factor FFT algorithm using distributed arithmetic," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 30, pp. 217–227, April 1982.
- [64] K. T. Chen, "A new record: the largest known prime number," *IEEE Spectrum*, vol. 27, p. 47, February 1990.
- [65] 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*, vol. 33, pp. 1231–1238, October 1985.
- [66] R. N. Bracewell, *The Hartley Transform*. Oxford Press, 1986.
- [67] F. M. Wang and P. Yip, "Fast prime factor decomposition algorithms for a family of discrete trigonometric transforms," *Circuits, Systems, and Signal Processing*, vol. 8, no. 4, pp. 401–419, 1989.
- [68] K. R. Rao and P. Yip, *Discrete Cosine Transform: Algorithms, Advantages, Applications*. San Diego, CA: Academic Press, 1990.
- [69] 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*, vol. 35, pp. 849–863, June 1987.
- [70] P. Duhamel and M. Vetterli, "Improved Fourier and Hartley transform algorithms, application to cyclic convolution of real data," *IEEE Trans. on ASSP*, vol. 35, pp. 818–824, June 1987.
- [71] M. Popović and D. Šević, "A new look at the comparison of the fast Hartley and Fourier transforms," *IEEE Transactions on Signal Processing*, vol. 42, pp. 2178–2182, August 1994.
- [72] P. R. Uniyal, "Transforming real-valued sequences: fast Fourier versus fast Hartley transform algorithms," *IEEE Transactions on Signal Processing*, vol. 42, pp. 3249–3254, November 1994.
- [73] 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*, vol. 45, pp. 2010–2022, August 1997.
- [74] G. Bruun, "Z-transform DFT filters and FFTs," *IEEE Transactions on ASSP*, vol. 26, pp. 56–63, February 1978.
- [75] R. Storn, "On the Bruun algorithm and its inverse," *Frequenz*, vol. 46, pp. 110–116, 1992. a German journal.
- [76] C. M. Rader and N. M. Brenner, "A new principle for fast Fourier transformation," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. ASSP-24, pp. 264–266, June 1976.
- [77] K. M. Cho and G. C. Temes, "Real-factor FFT algorithms," in *Proceedings of IEEE ICASSP-78*, (Tulsa, OK), pp. 634–637, April 1978.
- [78] P. Duhamel, B. Piron, and J. M. Etcheto, "On computing the inverse DFT," *IEEE Transactions on ASSP*, vol. 36, pp. 285–286, February 1978.
- [79] R. Singleton, "An algorithm for computing the mixed radix fast Fourier transform," *IEEE Transactions on Audio and Electroacoustics*, vol. AU-17, pp. 93–103, June 1969.
- [80] J. A. Glassman, "A generalization of the fast Fourier transform," *IEEE Transactions on Computers*, vol. C-19, pp. 105–116, February 1970.
- [81] W. E. Ferguson, Jr., "A simple derivation of Glassman general-n fast Fourier transform," *Comput. and Math. with Appls.*, vol. 8, no. 6, pp. 401–411, 1982. Also, in Report AD-A083811, NTIS, Dec. 1979.
- [82] L. Rabiner, R. Schafer, and C. Rader, "The chirp z-transform algorithm," *IEEE Transactions on Audio Electroacoustics*, vol. AU-17, pp. 86–92, June 1969.
- [83] G. A. Sitton, "The QFT algorithm," *unpublished paper*, 1985.
- [84] 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*, (IEEE ICASSP-94, Adelaide, Australia), pp. III:445–448, April 19–22 1994.
- [85] H. Guo, G. A. Sitton, and C. S. Burrus, "The quick Fourier transform, an FFT based on symmetries," *IEEE Transactions on Signal Processing*, to appear 1997.

- [86] 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*, (ISCAS-92, San Diego, CA), pp. 133–136, May 1992.
- [87] I. W. Selesnick and C. S. Burrus, "Multidimensional mapping techniques for convolution," in *Proceedings of the IEEE International Conference on Signal Processing*, (IEEE ICASSP-93, Minneapolis), pp. III–288–291, April 1993.
- [88] 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*, (IEEE ISCAS-94, London), pp. 2.449–452, June 30 1994.
- [89] I. W. Selesnick and C. S. Burrus, "Automatic generation of prime length FFT programs," *IEEE Transactions on Signal Processing*, vol. 44, pp. 14–24, January 1996.
- [90] W. K. Hocking, "Performing Fourier transforms on extremely long data streams," *Computers in Physics*, vol. 3, pp. 59–65, January 1989.
- [91] R. Stasiński, "Extending sizes of effective convolution algorithms," *Electronics Letters*, vol. 26, no. 19, pp. 1602–1604, 1990.
- [92] R. C. Agarwal and J. W. Cooley, "New algorithms for digital convolution," *IEEE Trans. on ASSP*, vol. 25, pp. 392–410, October 1977.
- [93] I. Pitas and C. S. Burrus, "Time and error analysis of digital convolution by rectangular transforms," *Signal Processing*, vol. 5, pp. 153–162, March 1983.
- [94] R. Meyer, R. Reng, and K. Schwarz, "Convolution algorithms on DSP processors," in *Proceedings of the ICASSP-91*, (Toronto, Canada), pp. 2193–2196, May 1991.
- [95] R. C. Agarwal and C. S. Burrus, "Number theoretic transforms to implement fast digital convolution," *Proceedings of the IEEE*, vol. 63, pp. 550–560, April 1975. also in IEEE Press DSP Reprints II, 1979.
- [96] 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*, (New York), pp. 1894–1897, April 1988.
- [97] 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*, (Espoo, Finland), pp. 1915–1918, June 1988.
- [98] C. Roche, "A split-radix partial input/output fast Fourier transform algorithm," *IEEE Transactions on Signal Processing*, vol. 40, pp. 1273–1276, May 1992.
- [99] 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*, vol. 41, pp. 1184–1200, March 1993.
- [100] A. R. Várkonyi-Kóczy, "A recursive fast Fourier transformation algorithm," *IEEE Transactions on Circuits and Systems, II*, vol. 42, pp. 614–616, September 1995.
- [101] R. Meyer and K. Schwarz, "FFT implementation on DSP-chips — theory and practice," in *Proceedings of the ICASSP-90*, (Albuquerque, NM), pp. 1503–1506, April 1990.
- [102] 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*, (Albuquerque, NM), pp. 1507–1510, April 1990.
- [103] 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*, vol. 9, no. 4, pp. 449–500, 1990.
- [104] C. Van Loan, *Matrix Frameworks for the Fast Fourier Transform*. Philadelphia, PA: SIAM, 1992.
- [105] R. Tolimieri, M. An, and C. Lu, *Mathematics of Multidimensional Fourier Transform Algorithms*. New York: Springer-Verlag, 1993. Second edition, 1997.
- [106] J. Granata, M. Conner, and R. Tolimieri, "The tensor product: a mathematical programming language for FFTs," *IEEE Signal Processing Magazine*, vol. 9, pp. 40–48, January 1992.
- [107] J. Granata, M. Conner, and R. Tolimieri, "Recursive fast algorithms and the role of the tensor product," *IEEE Transactions on Signal Processing*, vol. 40, pp. 2921–2930, December 1992.

- [108] H. Guo and C. S. Burrus, "Approximate FFT via the discrete wavelet transform," in *Proceedings of SPIE Conference 2825*, (Denver), August 6–9 1996.
- [109] H. Guo and C. S. Burrus, "Wavelet transform based fast approximate Fourier transform," in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, (IEEE ICASSP-97, Munich), pp. III:1973–1976, April 21–24 1997.
- [110] H. Guo and C. S. Burrus, "Fast approximate Fourier transform via wavelet transforms," *IEEE Transactions on Signal Processing*, submitted, January 1997.
- [111] C. S. Burrus, R. A. Gopinath, and H. Guo, *Introduction to Wavelets and the Wavelet Transform*. Upper Saddle River, NJ: Prentice Hall, 1998.

This is a note describing results on efficient algorithms to calculate the discrete Fourier transform (DFT). The purpose is to report work done at Rice University, but other contributions used by the DSP research group at Rice are also cited. Perhaps the most interesting is the discovery that the Cooley-Tukey FFT was described by Gauss in 1805 [1]. That gives some indication of the age of research on the topic, and the fact that a recently compiled bibliography [2] on efficient algorithms contains over 3400 entries indicates its volume. Three IEEE Press reprint books contain papers on the FFT