Mathematical Problems of Computer Science 33, 48-53, 2010.

Efficient Computation of Subset of Output Points of FFT

Rafayel Barseghyan

Institute for Informatics and Automation Problems of NAS RA

Abstract

This paper presents efficient pruning algorithms for computing the length $q \times 2^p$ - DFT for a subset of output points based on transform decomposition method and in new results in computation of FFT.

1. Introduction

The discrete Fourier transform (DFT) finds applications in almost every field of science and engineering. A major reason for its widespread use is the existence of efficient algorithms for its computation. Family of these algorithms are known as fast Fourier transforms (FFT). Cooley and Tukey [1] in 1965 suggested a new method for computation of the DFT which required only $O(N \log N)$ operations in opposite of direct form computation $O(N^2)$. Later become more efficent ways to compute DFT, such as prime factor algorithm(PFA) [2], Winograd Fourier transform algorithm (WFTA) [2], split-radix FFT and modified split-radix FFT algorithms [4],[5].There are many applications where only subset of output of FFT are needed. This problem can be solved by using Goertzel's algorithm [6], [2] which for computation one output point requires $(2N-1)$ real multiplications and $(4N-1)$ real additions for input vector in N-length. Another algorithm, which gives a slight improvement over Goertzels algorithm, was proposed by Boncelet [7].Boncelet's method requires $O(\frac{8N}{3})$ real operations per output point. More efficient methods to compute partial spectrum are called pruned FFT algorithms (some authors called the "zoom-FFT"). The pruning method was first proposed by Markel [9] in 1971. Later this method was improved by Skinner [10] (See Figure 1), Nagai [11]. Efficient method for solving this problem is proposed by Sorensen, Burrus and Jones [11] which is based on fast Fourier transform Decomposition (FFT TD). The main advantage of TD -method is in her universality, which allows to use the best of known FFT algorithms. Algorithm proposed in this paper uses modified split-radix FFT algorithms [4],[5],[11],[12].

2. Description of the algorithm

Let $x = (x_0, x_1, \ldots, x_{N-1})$ be an original column-vector with complex components (N is the power of two.$N = 2^m$). DFT of this vector is determined by the following formula

$$X[k] = \frac{1}{N}\sum_{n=0}^{N-1} x[n]W_N^{nk}, \qquad k = 0, 1, \ldots, N-1, \tag{1}$$

where $W_N = \exp(-j\frac{2\pi}{N}) = \cos\frac{2\pi}{N} - j\sin\frac{2\pi}{N}, \quad j = \sqrt{-1}$.
Below coefficient $\frac{1}{N}$ was omitted.

Let we only in P part of output spectrum $P = 2^l$. Introduce following notations

$$\begin{aligned} &Q = \tfrac{N}{P}, \\ &n = Qn_1 + Qn_2, \\ &n_1 = 0, 1 \ldots, P-1, \\ &n_2 = 0, 1 \ldots, Q-1. \end{aligned} \tag{2}$$

Now we can represent (1) as follows

$$X[k] = \sum_{n_2=0}^{Q-1}\sum_{n_1=0}^{P-1} x[n_1Q + n_2]W_N^{(n_1Q+n_2)k} \tag{3}$$

Rewrite (2) as follows

$$X[k] = \sum_{n_2=0}^{Q-1}(\sum_{n_1=0}^{P-1} x[n_1Q + n_2]W_P^{n_1<k>_P})W_N^{n_2k} \tag{4}$$

Here $<>_P$ denotes reduction modulo P and $k = 0.1, \ldots N-1$. Splitting (4) into two equations

$$X[k] = \sum_{n_2=0}^{Q-1} X_{n_2}[<k>_P]W_N^{n_2k} \tag{5}$$

and

$$X_{n_2}[j] = \sum_{n_1=0}^{P-1} x[n_1Q + n_2]W_P^{n_1j} = \sum_{n_1=0}^{P-1} xn_2[n_1]W_P^{n_1j}, \tag{6}$$

where $j = 0, 1, \ldots, P-1$ and

$$x_{n_2}[n_1] = x[n_1Q + n_2]. \tag{7}$$

The last part of (6) can be computed with any FFT program. Data which are interests us depend on n_2. DFT has to be computed for all values of n_2 and Q P-length FFTs. (Scheme of this method is demonstrated in Figure 2) Results of FFTs are restructured using (5), which can be evaluated by using $(Q-1)$- complex additions and Q-complex multiplications. We need to do PQ -complex multiplications (one complex multiplication requires 4 real multiplication and 2 real additions) and $P(Q-1)$ -complex additions (with 2 real additions) By representing (5) as follows

$$X[k] = X_0[<k>_P] + \sum_{n_2=1}^{Q-1} X_{n_2}[<k>_P]W_N^{n_2k} \tag{8}$$

we can save P- complex multiplications over PQ=N. Resuming these results can find the total complexity of pruned FFT algorithm with TD

$$C_P^{+}(TD) = Q \times C_P^{+}(FFT) + 4(N-P) \tag{9}$$

$$C_P^{\times}(TD) = Q \times C_P^{+}(FFT) + 4(N - P) \tag{10}$$

In [10] other technique similar to Goertzel's algorithm for receiving lower bounds in arithmetic complexity of (9) and (10) are described. Derivation of that method is not described in this paper, but complexity of algorithm is

$$C_P^{+}(G) = Q \times C_P^{+}(FFT) + 4P(Q - 1) + 4P \tag{11}$$

$$C_P^{\times}(G) = Q \times C_P^{+}(FFT) + 2P(Q - 1) + 4P \tag{12}$$

This universal approach allows us to use most efficient FFT algorithm for computing P-point DFT. For FFT algorithm scheme with 3 real multiplications and 3 real additions per complex multiplication are chosen[11]. Complexity of this algorithm is presented below

$$C_N^{+} = 3N \log_2 N - \frac{9}{2}N - 12(-1)^{\log_2 N} + 4 \tag{13}$$

$$C_N^{\times} = N \log_2 N - \frac{7}{2}N - 4(-1)^{\log_2 N} + 4 \tag{14}$$

By substituting (13) and (14) in (11) and (12) respectively we finally get

$$C_P^{+}(TD) = 3N \log_2 P - \frac{1}{2}N - 4(P - Q) - 12(-1)^{\log_2 P} \tag{15}$$

$$C_P^{\times}(TD) = N \log_2 P - \frac{1}{2}N - 4(P - Q) - 4Q(-1)^{\log_2 P} \tag{16}$$

and when method similar to Goertzel's algorithm is used

$$C_P^{+}(G) = 3N \log_2 P - \frac{1}{2}N - 4Q(3Q(-1)^{\log_2 N} - 1) \tag{17}$$

$$C_P^{\times}(G) = N \log_2 P - \frac{3}{2}N + 2(P + 2Q) - 4Q(-1)^{\log_2 N} \tag{18}$$

When complex multiplication requires 4 real multiplications and 2 real additions can be applied algorithm [4].

$$C_N^{+} = \frac{8}{3}N \log_2 N - \frac{16}{9}N - \frac{2}{9}(-1)^{\log_2 N} + 2 \tag{19}$$

$$C_N^{\times} = \frac{10}{9}N \log_2 N - \frac{76}{27}N - 2\log_2 N - \frac{2}{9}N(-1)^{\log_2 N} + \frac{22}{27}(-1)^{\log_2 N} + 6 \tag{20}$$

Given these results we finally find

$$C_P^{+}(TD) = \frac{8}{3}N\log_2 P - \frac{20}{9}N - \frac{2}{9}Q(-1)^{\log_2 P} - 2(2P - Q) \tag{21}$$

$$C_P^{\times}(TD) = \frac{10}{9}N \log_2 P + \frac{32}{27}N + 2(3Q - 2P) - [2\log_2 P + \frac{2}{9}P(-1)^{\log_2 P} - \frac{22}{27}(-1)^{\log_2 P}]Q \tag{22}$$

3. DFT of composite sequence lengths

In [13] the idea of split-radix algorithm has been extended to the length $q \times 2^m$ DFTs. Logarithmic complexity of algorithm for $q \times 2^m$ length DFTs doesn't exist. It is easy to see what arithmetic complexity of algorithm [13] is described by following recurrence relations. For complex multiplication with 4 real multiplications and 2 real additions

$$C_N^+ = C_{N/2}^+ + 2C_{N/4}^+ + 4N - 4q, \qquad C_N^\times = C_{N/2}^\times + 2C_{N/4}^\times + 2N - 12q \tag{23}$$

For complex multiplication with 3 real multiplications and 3 real additions

$$C_N^+ = C_{N/2}^+ + 2C_{N/4}^+ + \frac{9}{2}N - 8q, \qquad C_N^\times = C_{N/2}^\times + 2C_{N/4}^\times + \frac{3}{2}N - 8q \tag{24}$$

Using these initial conditions

$$\begin{aligned} C_{2q}^+ &= 2C_q^+ + 4q, \quad C_{4q}^+ = 4C_q^+ + 16q \\ C_{2q}^\times &= 2C_q^\times, \qquad C_{4q}^\times = 4C_q^\times \end{aligned} \tag{25}$$

Finally we get
for complex multiplication with 4 real multiplications and 2 real additions

$$\begin{aligned} C_N^+ &= \tfrac{8}{3}N\log_2\left(\tfrac{N}{q}\right) - N\left(\tfrac{16}{9} - \tfrac{1}{q}C_q^+\right) - \tfrac{2q}{9}(-1)^{\log_2\left(\frac{N}{q}\right)} + 2q \\ C_N^\times &= \tfrac{4}{3}N\log_2\left(\tfrac{N}{q}\right) - N\left(\tfrac{38}{9} - \tfrac{1}{q}C_q^\times\right) + \tfrac{2q}{9}(-1)^{\log_2\left(\frac{N}{q}\right)} + 6q \end{aligned} \tag{26}$$

for complex multiplication with 3 real multiplications and 3 real additions

$$\begin{aligned} C_N^+ &= 3N\log_2\left(\tfrac{N}{q}\right) - N\left(3 - \tfrac{1}{q}C_q^+\right) + 4q \\ C_N^\times &= N\log_2\left(\tfrac{N}{q}\right) - N\left(3 - \tfrac{1}{q}C_q^\times\right) + 4q \end{aligned} \tag{27}$$

Now by using (9) and (10) we can get
for complex multiplication with 4 real multiplications and 2 real additions

$$\begin{aligned} C_P^+(TD) &= \tfrac{8}{3}N\log_2\left(\tfrac{P}{q}\right) - N\left(\tfrac{16}{9} - \tfrac{1}{q}C_q^+\right) - \tfrac{2q}{9}Q(-1)^{\log_2\left(\frac{N}{q}\right)} + 2Qq + 4(N-P) \\ C_P^\times(TD) &= \tfrac{4}{3}N\log_2\left(\tfrac{P}{q}\right) - N\left(\tfrac{38}{9} - \tfrac{1}{q}C_q^\times\right) + \tfrac{2q}{9}Q(-1)^{\log_2\left(\frac{N}{q}\right)} + 6Qq + 4(N-P) \end{aligned} \tag{28}$$

for complex multiplication with 3 real multiplications and 3 real additions

$$\begin{aligned} C_P^+(TD) &= 3N\log_2\left(\tfrac{P}{q}\right) - N\left(3 - \tfrac{1}{q}C_q^+\right) + 4(N - P + Qq) \\ C_P^\times(TD) &= N\log_2\left(\tfrac{P}{q}\right) - N\left(3 - \tfrac{1}{q}C_q^\times\right) + 4(N - P + Qq) \end{aligned} \tag{29}$$

4. Conclusion

The problem of computation of subset of output points can be performed by three basic methods.

1. One can ignore the fact that subset of output points are need and compute all points.

This method often requires more operations that other methods, but it is easy to implement.
2. Can compute some points by using Goertzel's algorithm (or Boncelet), but this method efficient only for very small subset of output points.
3. Transform decomposition method
So we get

- Described methods are efficient if needed points count less than 50 percent of spectrum.
- If required points count are comparable to N savings of presented methods are minimal and if subset greater than 50 percent of spectrum it is more efficient to use efficient FFT algorithm.

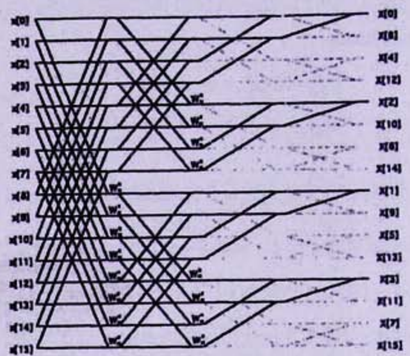

Figure 1.Modified Skinner's pruned 16-point FFT algorithm for 4-output points.

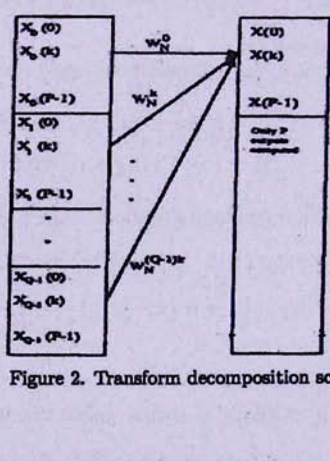

Figure 2. Transform decomposition scheme.

References

[1] J. W. Cooley and J. W. Tukey, "An algorithm for the machine computation of the complex Fourier series", *Math. Computation* , pp. 297–301, 1965.

[2] R. Blahut, Fast Algorithms for Digital Signal Processing. Addison-Wesley, 1985.

[3] R. Yavne, "An economical method for calculating the discrete Fourier transform," *Proc. AFIPS Fall Joint Computer Conf.*, pp. 115125, 1968 .

[4] M. Frigo and S. G. Johnson,"A modified split-radix FFT with fewer arithmetic operations", *IEEE Trans. Signal Proccessing*, - vol. 55, pp. 111-119, 2207.

[5] H. Sarukhanyan, R. Barseghyan, "New Efficient FFT with Fewer Operations", CSIT, pp. 351-354, 2009.

[6] G. Goertzel,"An Algorithm for the evaluation of finite trigonometric series", *Amer. Math. Monthly*, pp. 34-35, 1958.

[7] C. G. Boncelet. "A rearranged DFT algorithm requiring $N^2/6$ multiplications," *IEEE Trans. Acoust., Speech, Signal Processing* . pp. 1658-1659, 1986.

[8] R. de Wild, "Method for partial spectrum computation," *Proc. Inst. Elec. Eng.*, pp. 659-666, 1987.

[9] J.D. Markel, "FFT pruning," *IEEE Trans. Audio Electroacoust*, pp. 305-311, 1971.

[10] H. V. Sorensen, C. S. Burrus, and D. L. Jones. "A new efficient algorithm for computing a few DFT points," *Proc. IEEE Int.*, pp. 1915-1918, 1988.

[11] R. Barseghyan,"Split-radix FFT algorithm with lifting steps," *Vestnik RAU, Series: Physics-mathematics and natural science* pp. 41-51, 2009.

[12] H. Sarukhanyan, R. Barseghyan."New Approach to FFT algorithms", *Mathematical Problems of Computer Science*vol. 33. pp. 24-34, 2010.

[13] G. Bi and Y. Q. Chen, "Fast DFT algorithms for length ," *IEEE Trans. Circuits and Syst.- II: Analog and Digital Signal Processing*, pp. 685-690, 1998.

ՖԱՁ ելքային որոշակի տիրույթի հաշվման արդյունավետ ալգորիթմ

Ռ. Բարսեղյան

Ամփոփում

Ձևափոխության դեկոմպոզիցիայի հիման վրա աշխատանքում մշակված է սպեկտրալ տիրույթի որոշակի հատվածի հաշվման արդյունավետ ալգորիթմ: