Novel Algorithms for the Accurate, Efficient, and Parallel Computation of Multidimensional, Regional Discrete Fourier Transforms

Marios S. Pattichis

Abstract–A novel class of Discrete Fourier Transform algorithms is presented. First, a new algorithm is presented for computing the DFT spectrum along any given direction. Then, computation of the entire DFT spectrum is computed using a minimal set of independent directions. It is shown that the new class of algorithms is both faster and more accurate than the traditional tensor decomposition of the DFT computation. Furthermore, the new algorithms allow for each direction to be computed independently of the others, hence allowing a parallel implementation.

I. Introduction

We express the m-dimensional DFT as:

$$X(k_1,\ldots,k_m) = \sum_{n_1=0}^{n_1=N_1-1}\cdots\sum_{n_m=0}^{n_m=N_M-1} x(n_1,\ldots,n_m)\cdot W_N^{k_1n_1+\cdots+k_mn_m} \quad (1)$$

where:

$$W_N = \exp\left[-\sqrt{-1}\frac{2\pi}{N}\right].$$

We are interested in computing $X(k_1,\ldots,k_m)$ for certain discrete frequencies $(k_1,\ldots,k_m)\in S$. We will consider the cases when: (i) S contains a single discrete frequency, (ii) S contains several discrete frequencies within a limited set of directions, and (iii) S contains all possible discrete frequencies. In the third case, (1) reduces to the problem of computing the m-dimensional DFT. More general regions of S may also be computed by combining the algorithms from any of the three cases that are described.

To motivate the new approach, we review the standard two-dimensional decomposition of the algorithm [1, 2, 3, 4]:

1. Compute the FFT along each row.
2. Transpose.
3. Compute the FFT along each column.
4. Transpose.

The important limitations of this approach are:

- For each DFT frequency, we require the accuracy computation of $N+1$ FFTs, limiting accuracy.
- Due to the transpositions, there is a lot of required communications among the processes, hence limiting the degree of parallelism, and computational efficiency.
- There are no efficient algorithms for handling sparse images in any finite number of dimensions.

By breaking DFT computation into a collection of one-directional FFTs, it is possible to overcome some of these difficulties by:

- Requiring a single FFT for each DFT frequency, hence yielding more accuracy.
- Avoiding transpositions altogether, and communicating only the data sums for each root of unity, hence reducing the data communication drastically for signals of any finite dimension.
- Allows for the efficient DFT computation along particular directions of the spectrum, as well as for sparse images, by considering the non-zero data samples for each root of unity.

In Section II, we introduce the basic ideas by looking at the problem of DFT computation for a 4×4 array. In Section III, the algorithm for computing the DFT spectrum along any given direction is presented. It is built upon an algorithm for computing a single point in the spectrum. It is shown that it requires a single, one-dimensional FFT. In Section IV, two minimal sets of independent directions for covering the 2-D and 3-D DFT frequency planes are computed. By using the smallest number of independent directions, it is shown that the 2-D and 3-D algorithms require a smaller number of 1-D FFTs than the popular tensor decomposition of the algorithm. In Section V, a summary of future work is presented.

II. Two-Dimensional DFT Examples

In this Section, we introduce the algorithm for computing the DFT spectrum along a particular direction. The method is similar to the slice DFT algorithm [4]. Nevertheless, the algorithms presented here are more advanced in that: (i) they employ SIMD additions to exploit modern microprocessor architectures, and (ii) access the memory sequentially and hence avoiding Cache misses.

Marios S. Pattichis is with the Department of Electrical and Computer Engineering, and the Center of High Performance Computing of the University of New Mexico, Albuquerque, NM 87131.

0-7803-6290-X/00/$10.00 ©2000 IEEE

Consider the 4×4 case:

$$X(k_1,k_2) = \sum_{n_1=0}^{n_1=3}\sum_{n_2=0}^{n_2=3} x(n_1,n_2)\cdot W_4^{k_1n_1+k_2n_2}$$

Consider the case for 4×4 where the 2-d array is stored in a row by row order. The 2-d 4×4 array corresponds to the following 1-d offsets:

$$\begin{bmatrix} x(0) & x(1) & x(2) & x(3) \\ x(4) & x(5) & x(6) & x(7) \\ x(8) & x(9) & x(10) & x(11) \\ x(12) & x(13) & x(14) & x(15) \end{bmatrix}$$

In what follows, we will use the fact that each row is made up of consecutive entries, but we could relax the fact that the array entries of row $n+1$ immediately follow the entries for row n, and the algorithms will still hold.

For the horizontal frequencies (see Figure 1):

$$\begin{bmatrix} S(0) \\ S(1) \\ S(2) \\ S(3) \end{bmatrix} = \begin{bmatrix} x(0) \\ x(1) \\ x(2) \\ x(3) \end{bmatrix} + \begin{bmatrix} x(4) \\ x(5) \\ x(6) \\ x(7) \end{bmatrix} + \begin{bmatrix} x(8) \\ x(9) \\ x(10) \\ x(11) \end{bmatrix} + \begin{bmatrix} x(12) \\ x(13) \\ x(14) \\ x(15) \end{bmatrix}$$

Note that all accesses to memory are sequential with stride=1. Also, the additions can be performed using SIMD additions for adding groups of four numbers:

$$[s(0:3)] = A_4 \begin{bmatrix} A_4 & & \\ & A_4 & \\ & & A_4 \end{bmatrix} \begin{bmatrix} x(0:3) \\ x(4:7) \\ x(8:11) \\ x(12:15) \end{bmatrix}$$

The horizontal frequencies are then obtained using a one-dimensional, 4-point DFT:

$$X(0,k) = \sum_{n=0}^{n=3} S(n)\cdot W_4^{nk}$$

Similarly, for the vertical frequencies (see Figure 1):

$$S(0) = x(0)+x(1)+x(2)+x(3)$$
$$S(1) = x(4)+x(5)+x(6)+x(7)$$
$$S(2) = x(8)+x(9)+x(10)+x(11)$$
$$S(3) = x(12)+x(13)+x(14)+x(15)$$

which again access the memory sequentially. Each sum can be computed using (for example):

$$S(0) = A_2 \begin{bmatrix} A_2 & \\ & A_2 \end{bmatrix} \begin{bmatrix} x(0) \\ x(1) \\ x(2) \\ x(3) \end{bmatrix}$$

Then, the vertical frequencies are computed using a one-dimensional DFT:

$$X(k,0) = \sum_{n=0}^{n=3} S(n)\cdot W_4^{nk}$$

For an arbitrary direction, the roots of unity may not be aligned as given for the horizontal and vertical frequencies. Nevertheless, it is possible to access the memory sequentially and align the entries between the rows. This is illustrated for the diagonal frequencies (see Figure 1). We compute:

$$S(0) = x(0)+x(7)+x(10)+x(13)$$
$$S(1) = x(1)+x(4)+x(11)+x(14)$$
$$S(2) = x(2)+x(5)+x(8)+x(15)$$
$$S(3) = x(3)+x(6)+x(9)+x(12)$$

by using right rotations along each row:

$$r(0:3) = R_{1,4}x(4:7)$$
$$r(4:7) = R_{2,4}x(8:11)$$
$$r(8:11) = R_{3,4}x(12:15)$$

hence allowing SIMD additions for every 4 elements:

$$[s(0:3)] = A_4 \begin{bmatrix} A_4 & & \\ & A_4 & \\ & & A_4 \end{bmatrix} \begin{bmatrix} x(0:3) \\ r(0:3) \\ r(4:7) \\ r(8:11) \end{bmatrix}$$

All the diagonal frequencies are then computed using a one-dimensional DFT:

$$X(k,k) = \sum_{k=0}^{k=3} s(n)\cdot W_4^{nk}$$

III. DFT Computation Along ANY DIRECTION

In this Section, many of the results illustrated in Section II will be generalized to an arbitrary direction. Much of the mathematics is similar to the development of Orthogonal FM transforms that can be found in [5].

Suppose that:

$$N_1 = 2^{d_1}, N_2 = 2^{d_2}, \ldots, N_M = 2^{d_M}$$

Then, define:

$$G_r = \left\{ W_N^{r\cdot n} \mid r = 0, 1, \ldots, N-1 \right\}$$

where:

$$N = \max_i N_i .$$

We note that the roots of unity occur with equal multiplicity:

Theorem. In each DFT basis function:

$$W_{N_1}^{k_1\cdot n_1} \cdots W_{N_M}^{k_M\cdot n_M}, \text{ for } 0 \le n_i \le N_i,$$

each root of unity:

$$W_{N_1}^{k_1\cdot n_1} \cdots W_{N_M}^{k_M\cdot n_M} = W_N^r$$

occurs with equal multiplicity, regardless of the value of r.

Proof: The proof is fairly straight forwarded and omitted.

Theorem. Define $N = \max(N_1, N_2, \ldots, N_M)$. The roots of unity for a DFT basis function are given by: $t_{\mathbf{k}}(Q) = G_r$ where Q denotes the set of image points, $r = \gcd(\mathbf{k}') \bmod N$, and

$$\mathbf{k}' = \left(\frac{N}{N_1} k_1, \frac{N}{N_2} k_2, \ldots, \frac{N}{N_M} k_M \right).$$

Proof:

From the properties of the $\gcd(\cdot)$, we can always find $\mathbf{n} \in Z^M$ such that $\gcd(\mathbf{k}') = \mathbf{n} \cdot \mathbf{k}'$. Then, we define $r = \gcd(\mathbf{k}') \bmod N$, and it follows that:

$$\begin{aligned} t_{\mathbf{k}}(\mathbf{n}) &= \exp\left[-j \frac{2\pi}{N_1} (k_1 n_1) \right] \cdots \exp\left[-j \frac{2\pi}{N_M} (k_M n_M) \right] \\ &= \exp\left[-j \frac{2\pi}{N} (\mathbf{n} \cdot \mathbf{k}') \right] \\ &= W_N^r \end{aligned}$$

It is also clear that:

$$W_N^{r \cdot d} = t_{\mathbf{k}}(d \cdot \mathbf{n}),$$

and hence that $G_r \subseteq t_{\mathbf{k}}(Q)$.

Since $r = \gcd(\mathbf{k}') \bmod N$, we can also find an integer a such that $r + Na = \gcd(\mathbf{k}')$. Define a new vector $\mathbf{p}$ by factoring out $\gcd(\mathbf{k}')$ from each component of $\mathbf{k}'$:

$$p_i = \frac{k_i'}{(r + Na)}$$

so that:

$$\mathbf{k}' = (r + Na)\mathbf{p}$$

For any $\mathbf{m} \in Q$, we have:

$$\begin{aligned} t_{\mathbf{k}}(\mathbf{m}) &= \exp\left[-j \frac{2\pi}{N_1} (k_1 m_1) \right] \cdots \exp\left[-j \frac{2\pi}{N_M} (k_M m_M) \right] \\ &= \exp\left[-j \frac{2\pi}{N} (\mathbf{m} \cdot \mathbf{k}') \right] \\ &= \exp\left[-j \frac{2\pi}{N} \mathbf{m} \cdot (r + Na)\mathbf{p} \right] \\ &= W_N^{r(\mathbf{m} \cdot \mathbf{p})} \end{aligned}$$

It is clear that $W_N^{r(\mathbf{m} \cdot \mathbf{p})} \in G_r$, and this implies that: $t_{\mathbf{k}}(Q) \subseteq G_r$.

Q.E.D.

Using the previous theorem, we rewrite (1) as:

$$X(\mathrm{k}) = \sum_{z \in t_{\mathbf{k}}(Q)} \left(\sum_{\mathbf{n} \in t_{\mathbf{k}}^{-1}(z)} x(\mathbf{n}) \right) z$$

This equation describes a simple algorithm for computing a single DFT coefficient. We simply add up all the signal points that "see" the same root of unity, multiply by that particular root of unity, and then sum over all of them.

Most importantly, the algorithm can be generalized for an arbitrary direction using:

$$y(i) = \sum_{\mathbf{n} \in t_{\mathbf{k}}^{-1}(z)} x(\mathbf{n}), \quad z = W_N^{i \cdot r}$$

and:

$$X(m\mathbf{k}) = \sum_{i=0}^{i = N / \gcd(r, N)} y(i) W_N^{m(i \cdot r)}$$

The last equation corresponds to a one-dimensional DFT.

IV. Results In Two and Three Dimensions

It is interesting to generalize the directional DFT algorithm for computing the entire DFT spectrum. This is done in this Section where minimal sets of directions for covering the 2-D and 3-D frequency planes are computed.

To cover the frequency plane S, define:

$$V = \{ \mathbf{v} \in S \mid \exists i \text{ such that } : v_i = 1 \}.$$

We next define what we mean by "covering".

Definition 1. A set S' is said to **cover** S if for every $\mathbf{v} \in S$, we can find a $\mathbf{v}' \in S'$, and a non-negative integer a, such that $\mathbf{v} = a\mathbf{v}'$.

Theorem 1. Let $\mathbf{x} \in S$, where each $N_i = 2^{p_i}$ for p_i being a positive integer. Then $\mathbf{x}$ can be expressed as $\mathbf{x} = a\mathbf{v}$ for some $\mathbf{v} \in V$, $a \geq 0$ being a non-negative integer. In other words, we say that V covers S.

Proof

There are two cases to consider:

1. For some x_i component of $\mathbf{x}$, we have that $\gcd(x_i, 2^{p_1 + p_2 + \cdots + p_M}) = 1$.
2. For all x_i components of $\mathbf{x}$, we have that $\gcd(x_i, 2^{p_1 + p_2 + \cdots + p_M}) \neq 1$.

In the first case, it is clear that x_i does not contain a power of two in its prime number decomposition, and hence $\gcd(x_i, 2^{p_j}) = 1$ for $0 \leq j \leq N_j - 1$. Let $\mathbf{v} = (v_1, v_{i-1}, 1, v_{i+1}, \ldots, v_M)$ be arbitrary. Then $x_i \mathbf{v} = (x_i v_1, x_i v_2, \ldots, x_i v_{i-1}, x_i, x_i v_{i+1}, \ldots, x_i v_M)$. Since $\gcd(x_i, 2^{p_j}) = 1$ for $0 \leq j \leq N_j - 1$, it is clear

that there exist unique d_j such that $x_i d_j = x_j \pmod{N_j}$ for $0 \le j < i$, $i < j \le M$. Next, we set $v_j = d_j$ for $0 \le j < i$, $i < j \le M$, and observe that $x_i \mathbf{v} \doteq \mathbf{x}$ and $\mathbf{v} \in V$.

In the second case, all elements of $\mathbf{x}$ are powers of two. Let x_j be the smallest element of $\mathbf{x}$. It is clear that

$x_j \mathbf{v} = \mathbf{x}$ where

$\mathbf{v} = \left(2^{q_1}, 2^{q_2}, \ldots, 2^{q_{j-1}}, 1, 2^{q_{j+1}}, \ldots, 2^{q_M}\right)$ and

$2^{q_i} = x_i / x_j$. Clearly $\mathbf{v} \in V$.

QED

It is possible to cover V (and hence S), using one of its subsets: $V_C \subseteq V$. In two-dimensions, define V_C by:

$$V_1 = \{(i,1) \mid i = 0, 1, \ldots, N_1 - 1\}$$
$$V_2 = \{(1,i) \mid i = 0, 2, 4, \ldots, N_2 - 1\}$$
$$V_C = V_1 \cup V_2$$

It is noted that in V_C, there are only $N_1 + N_2/2$ directions. In turn, these correspond to $N_1 + N_2./2$ one-dimensional FFTs as opposed to $N_1 + N_2$ one-dimensional FFTs for the standard 2-d algorithm.

In three-dimensions, define V_C by:

$$V_1 = \{(1,i,j) \mid 0 \le i \le N_2 - 1, 0 \le j \le N_3 - 1\}$$
$$V_2 = \{(j,i,1) \mid j = 0, 2, 4, \ldots, N_1 - 1, 0 \le i \le N_2 - 1\}$$
$$V_3 = \{(i,j,1) \mid i = 0, 2, \ldots, N_1 - 1, j = 0, 2, \ldots, N_2 - 1\}$$
$$V_C = V_1 \cup V_2 \cup V_3$$

For a three-dimensional image of size $N \times N \times N$, it is noted that the set includes only $7N^2/4$ independent directions. In turn, this requires $7N^2/4$ 1-d FFTs, a significant reduction over the $3N^2$ 1-d FFTs required by the standard 3-d algorithm.

Perhaps most importantly is the following theorem that states that the numbers of directions are minimal:

Theorem 3. The cardinality of both the 2-d and 3-d V_C, $|V_C|$ are minimal in the sense that there does not exist another set V_C' that covers S, yet satisfying $|V_C| > |V_C'|$.

The complete proof of this theorem is unfortunately too long to be included in this paper. Briefly though, in both the 2-D and 3-D sets, at-least one vector component will only assume only even integer values, and hence avoid the unit component of the other vectors.

The minimal sets can also be extended to higher dimensions.

V. Conclusion and Future Work

A novel class of DFT computation algorithms have been presented. In the near future, paralllel algorithms will be implemented at the Albuquerque High Performance Computing Center.

VI. Referernces

[1] A. Antoniou, *Digital Filters: Analysis and Design*, McGraw-Hill, TMH Edition, New Delhi, 1988.

[2] A.V. Oppenheim, R.W. Schafer, with J.R. Buck, *Discrete-Time Signal Processing,*

[3] C.V. Loan, *Computational Frameworks for the Fast Fourier Transform*, SIAM Frontiers in Applied Mathematics, Philadelphia, 1992.

[4] D.E. Dudgeon and R.M. Mersereau, *Multidimensional Digital Signal Processing*, Prentice-Hall, New Jersey, 1984.

[5] M.S. Pattichis, "AM-FM Transforms with Applications." Ph.D. diss., The University of Texas at Austin, 1998.

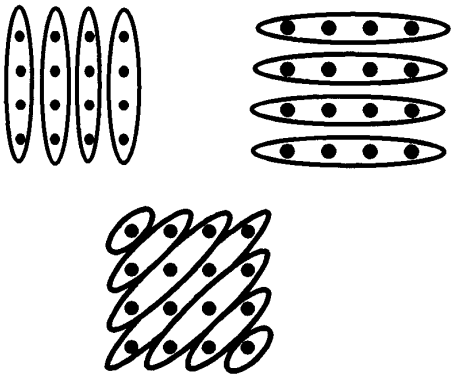

Figure 1. From top left to bottom center, distribution of the roots of unity for: (i) horizontal frequencies, (ii) vertical frequencies and (iii) diagonal frequencies. Each ellipse contains the discrete points that share the same root of unity. For the diagonal frequencies, we have a leftwise rotation from the roots of unity of each row to the row below (see text for details). This allows us to rotate the image samples along each row in order to add them to the row above. This pattern also exists along rows of any finite dimension.