A sparse data fast fourier transform (SDFFT)

4 authors, including:

Alaeddin Aydiner
Intel
36 PUBLICATIONS   478 CITATIONS

SEE PROFILE

Weng Cho Chew
University of Illinois, Urbana-Champaign
358 PUBLICATIONS   8,368 CITATIONS

SEE PROFILE

Jiming Song
Iowa State University
222 PUBLICATIONS   7,783 CITATIONS

SEE PROFILE

Some of the authors of this publication are also working on these related projects:

Project   Manipulating Linear and Nonlinear Responses from Two-Dimensional Materials View project

Project   Inverse scattering View project

A Sparse Data Fast Fourier Transform (SDFFT) – Algorithm and Implementation*

Alaeddin A. Aydıner, Weng Cho Chew, Jiming Song

Center for Computational Electromagnetics

Department of Electrical and Computer Engineering

University of Illinois at Urbana-Champaign

Abstract

An algorithm that efficiently Fourier transforms sparse spatial data to sparse spectral data with controllable error is presented. Unlike the ordinary nonuniform fast Fourier transform (NUFFT), which becomes $O(N^2)$ for sparse r-space *and* sparse k-space data, the sparse data fast Fourier transform (SDFFT) presented herein decreases the cost to $O(N \log N)$ while preserving the $O(N \log N)$ memory complexity. The algorithm can be readily employed in general signal processing applications where only part of the k-space is to be computed – regardless of whether it is a regular region like an angular section of the Ewald's sphere or it consists completely of arbitrary points. Among its applications in electromagnetics are back-projection tomography, diffraction tomography, synthetic aperture radar imaging, and the computation of far field patterns due to general aperture antennas and antenna arrays.

I Introduction

Various NUFFT algorithms have been utilized to extend the FFT algorithm to nonuniform spatial data [1-4]. However, these algorithms are not efficient when both the given spatial input and the desired spectral data are sparse in 2D or 3D. If the number of r-space and k-space points are denoted by $N_r \approx N$ and $N_k \approx N$ respectively, it can be readily seen that the complexity is $O(N_k N_r \log N_r) = O(N^2)$. Yet, the need for such a fast Fourier transform often arises in electromagnetics and signal processing. For instance, only a small angular section of the Ewald's sphere is relevant in the computation of the beamwidth between first nulls of a high-directivity antenna like the parabolic reflector.

The organization of this paper is as follows. In Section II, the algorithm is explained together with error controllability. Section III briefly discusses our implementation of the algorithm. Finally, conclusions and future work are discussed in Section IV.

II Algorithm

The multilevel fast multipole algorithm (MLFMA) [5-6] suggests a multilevel way of approaching the nonuniform FFT problem in which the lower levels have a smaller bandwidth in the k-space. The following algorithm is conceptually akin to the MLFMA which consists of aggregation, translation and disaggregation stages and involves a relatively complicated translation operator and interpolation between levels. In the case of the Fourier transform, the translation operator happens to have the simple form of a phase delay and there is no disaggregation stage.

The essence of the SDFFT algorithm is to express the transform in a multilevel fashion as detailed in [7]. We can hierarchically express the Fourier transform of f, vis., $F(\mathbf{k}_i) = \sum_{j=1}^{N_r} f(\mathbf{r}_j) e^{i\mathbf{k}_i \cdot \mathbf{r}_j}$ where $\mathbf{r}_j$ and $\mathbf{k}_i$ belong to nonuniform, arbitrary data in the r-space and k-space respectively by letting $\mathbf{r}_j = \mathbf{r}_{j1} + \mathbf{r}_{j2} + \ldots \mathbf{r}_{jn} + \Delta\mathbf{r}_j$. Here, the $\mathbf{r}_i$'s are vectors from the center of boxes at progressively deeper levels. The key observation is that the deepest-level contribution $\Delta\mathbf{r}_j$ has a small bandwidth in the k-space because Δr_j is small. Thus, by the Nyquist criterion $\Delta r \, \Delta k \leq \pi$, it is possible to interpolate such contributions very

sparsely in the k-space. This process can then be repeated by aggregating yet another level in the r-space, i.e., taking into account the next r-space vector and then interpolating to a finer grid covering the desired k-space data as dictated by the Nyquist condition.

Thus, the algorithm descends in the k-space as many levels as it ascends in the r-space, namely $O(\log N_r)$ levels. Also, the process starts with N_r r-space data mapped to $O(1)$ k-space points and ends with N_k k-space data interpolated from $O(1)$ r-space data. Hence, the computational complexity is $O((N_r + N_k) \log N_r)$ [7]. Because the workload at each level is about the same, the algorithm also preserves the $O(N \log N)$ memory complexity.

Provided the Nyquist condition is adequately satisfied, the interpolation can be performed with exponential accuracy since the exponential function is an entire function, i.e., it is infinitely smooth . Unlike the perfect sinc interpolation, all practical interpolation methods lead to an excess bandwidth which needs to be addressed by proper amount of oversampling.

There are essentially three ways of controlling the error of this algorithm: (a) increasing the spectral level such that Δk becomes smaller which is equivalent to satisfying the Nyquist criterion $\Delta r\, \Delta k \leq \pi$ better, (b) increasing the number of interpolation points, and (c) increasing the number of spatial levels which has the same effect as the first.

The first method has an associated memory cost because the next-to-last r-space level can be rather crowded and the number of interpolation nodes increases exponentially with dimension. Indeed, if the number of interpolation nodes per dimension is n_i, the total number of interpolation nodes is given by n_i^N where N is the dimension of the k-space tree. The second method has the problem that most interpolation schemes have a complexity higher than $O(N)$. The last method is the most cost-effective of all since it has a logarithmic cost, yet in most cases, it requires remeshing.
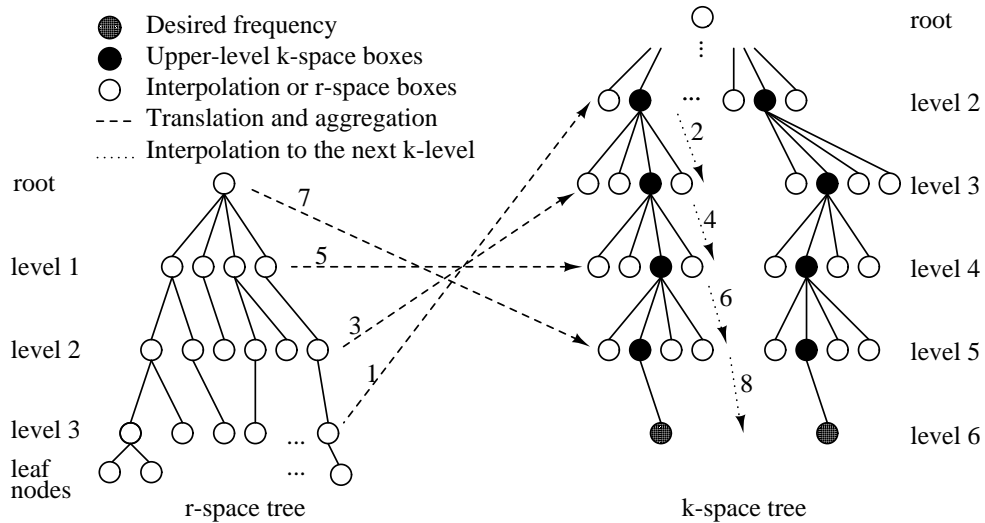
III   Implementation

Figure 1: An implementation of the algorithm using two trees. The 2D spatial and spectral trees each can have at most $2^{N=2} = 4$ branches. The numbers represent the sequence of operations.

Our implementation of the above algorithm involves two possibly unbalanced trees for the two domains as depicted in Figure 1. In the k-space tree, the SDFFT process starts at a level governed by the Nyquist

criterion. Since as Δr becomes larger, Δk becomes accordingly smaller, it suffices to check the Nyquist rate once at the beginning to determine the starting k-space level.

Thus far, we have employed Lagrange interpolation in the implementation. Error control by employing extra interpolation nodes is illustrated in Figure 2. The results are due to 6 levels of interpolation in a 2D k-space tree. Here, four interpolation nodes per dimension is the practical minimum admitted by the Nyquist condition. Unlike the case shown in Figure 1, some of the interpolation nodes may be common if the desired frequencies are sufficiently close. In such a case, SDFFT will perform better in terms of both storage and cpu time because a smaller k-space tree will need to be traversed.
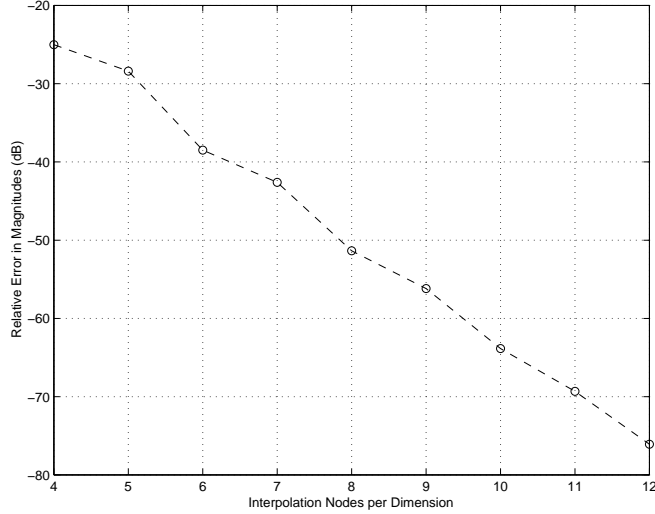

Figure 2: The error decreases exponentially with more interpolation nodes. These results are from a 2D problem.

The $O(N \log N)$ complexity of the algorithm is illustrated in Figure 3. Although it is a 1D simulation, the $N_k N_r$ product can easily become prohibitively large for $N = N_r = N_k \gg 1$. The lowest figure demonstrates that the average number of distinct interpolation nodes for each leaf is close to four, i.e., the pseudo-randomly generated k-vectors are sparse.

IV   Conclusions

- A sparse data fast Fourier transform (SDFFT) algorithm that can efficiently map sparse spatial data to sparse points in the spectral domain has been presented.

- The $O(N \log N)$ complexity of the algorithm as well as its error controllability are demonstrated.

  Currently, the algorithm is oriented more towards computing arbitrary points in the k-domain. Specializations of this algorithm for continuous curves in 2D or surfaces in 3D can be developed to alleviate the extra cost of generating a cover-set for the associated k-space data.

  Our future work will involve accelerating the current code, improving interpolation complexity, and applying the method to solving large-scale integral equations and inverse problems.
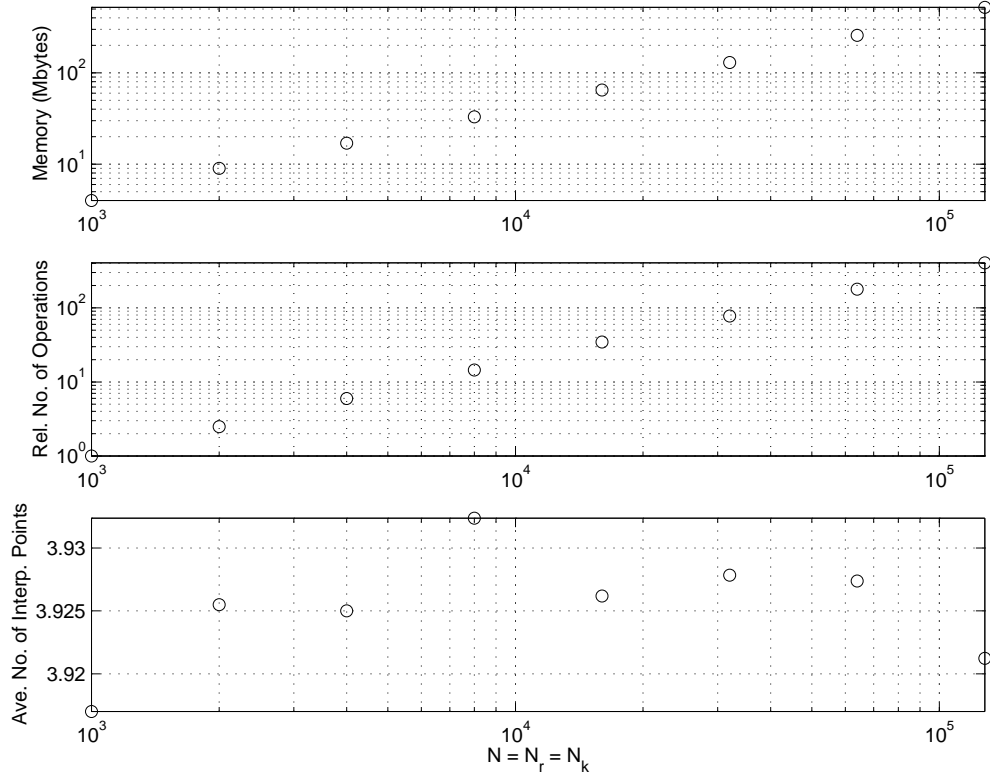
3

Figure 3: The uppermost two plots demonstrate the $O(N \log N)$ complexity of the algorithm. In the second plot, the relative number of operations are given as a function of N. The lowest figure is the average number of distinct interpolation nodes for each leaf node in the k-space tree.

References

[1] A. Dutt and V. Rokhlin, "Fast Fourier transforms for nonequispaced data," *SIAM Journal on Scientific Computing*, vol. 14, pp. 1368–1393, Nov. 1993.

[2] A. Dutt and V. Rokhlin, "Fast Fourier transforms for nonequispaced data, II," *Applied and Computational Harmonic Analysis*, vol. 2, pp. 85–100, Jan. 1995.

[3] G. Beylkin, "On the fast Fourier transform of functions with singularities," *Appl. Computat. Harmonic Anal.*, vol. 2, pp. 363–382, 1995.

[4] Q. Liu, X. Xu, B. Tian, and Z. Zhang, "Applications of nonuniform fast transform algorithms in numerical solutions of differential and integral equations," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 38, pp. 1551–1560, July 2000.

[5] C. C. Lu and W. C. Chew, "Fast far-field approximation for calculating the RCS of large objects," *Microwave and Optical Technology Letters*, vol. 8, pp. 238–241, Apr. 1995.

[6] J. M. Song and W. C. Chew, "Multilevel fast-multipole algorithm for solving combined field integral equations of electromagnetic scattering," *IEEE Transactions on Microwave Theory and Techniques*, vol. 10, pp. 14–19, Sept. 1995.

[7] W. C. Chew and J. M. Song, "Fast Fourier transform of sparse spatial data to sparse fourier data," in *IEEE Antennas and Propagation International Symposium*, vol. 4, pp. 2324–2327, July 2000.