Parallelization of NUFFT with Radial Data on Multicore Processors

Nikos Pitsianis and Xiaobai Sun
Department of Computer Science
Duke University, Durham NC 27707

Introduction

We introduce a parallelization scheme for rapid and high-resolution image formation via non-uniform fast Fourier transform (NUFFT) with radial data in the frequency domain, utilizing modern multicore processors. Frequency data sampled in radial fashion are seen in certain synthetic aperture radar systems and medical imaging systems. In a typical NUFFT algorithm for image formation, the conventional fast Fourier transform (FFT) is preceded by a convolution step that translates the frequency samples to the Cartesian grid points designated for the FFT. With respect to any specified accuracy requirement, the translation step exploits the locality in spatial support of chosen convolution kernel functions so that the number of arithmetic operations in sample translation is linearly proportional to the size of the sample ensemble. The parallelization scheme consists of data partition, execution scheduling and mutual exclusion rules. It is implemented on two modern multicore processors. We discuss also on alternative parallelization strategies.

Algorithm description

We may describe a forward discrete Fourier transform (DFT) as follows,

$$\mathbf{v}(T) := \mathbf{F}(T,S)\,\mathbf{u}(S), \tag{1}$$

where $\mathbf{u}(S)$ denotes the values of the source data sampled over the point set S; $\mathbf{v}(T)$, the target data evaluated over the point set T; and $\mathbf{F}(T,S)$, the DFT matrix specific to S and T. The values in $\mathbf{u}$ may depend on both measurements and numerical quadrature weights. Unlike the inverse DFT, the inverse NUDFT may be carried out by an iterative process with a sequence of forward NUDFTs. We focus therefore on the forward evaluation of (1).

A NUFFT algorithm reduces the arithmetic complexity in direct evaluation of (1) to the same order as the FFT, by an approximate factorization with sparse or structured matrix factors, see the seminal papers in [1] and [2] for the principle idea and analysis. For image formation the approximate factorization may be expressed as follows,

$$\begin{gathered}\mathbf{c}(T)\odot\mathbf{v}(T) = \mathbf{F}(T,\tilde{T})\cdot\mathbf{C}(\tilde{T},S)\cdot\mathbf{u}(S)+\mathbf{E},\\ \max_{t\in T,s\in S}|E(t,s)|<\epsilon,\end{gathered} \tag{2}$$

where $\mathbf{C}(\bar{S},S)\mathbf{u}(S)$ are the data translated from the sampling points in S to $\bar{S}$ (the Cartesian grid points in the frequency domain), $\mathbf{F}(T,\bar{S})$ is the conventional DFT matrix of size $|T|=|\bar{S}|$, $\odot$ denotes elementwise multiplication or scaling, and $\mathbf{c}(T)$ is the nonzero scaling function $\mathbf{c}(t)$ over T, which corresponds to the convolution kernel function $\mathbf{C}(t,s)$ over $S\times\bar{S}$. It is determined that $|\bar{S}|=\alpha|S|$ with $\alpha>1$ as the oversampling factor, which is often less than 2 for three-dimensional sample translation.

Besides the well-understood structures in the scaling and in the DFT, the translation matrix factor $\mathbf{C}(\bar{S},S)$ is sparse. Every column has at most w_c^3 nonzero elements, with three-dimensional data. A convolution kernel function $\mathbf{C}(t,s)$ with local support of diameter w_c is chosen to meet the accuracy requirement specified by ϵ, independent of the size $|S|$ or $|T|$. Therefore, the total number of arithmetic operations for the sample translation is linearly proportional to $|S|$, with to ϵ, and the complexity for an entire NUFFT is $O(|T|\log(|T|))+O(|S|)$.

We describe next the datum sets in the image formation problem under consideration. The image in spa-

tial domain is formed in voxels, on equally spaced locations in Cartesian coordinates. The frequency data are acquired in radials. The samples along a ray between the origin and an end point on the sphere are equally spaced. The trajectory of the end points is specified.

Parallelization

With the FFT made available on many existing multicore processors, the parallelization of NUFFTs is primarily on the sample translation, $\mathbf{u}(\bar{S}) := \mathbf{C}(\bar{S}, S)\,\mathbf{u}(S)$. The platform for parallel NUFFT is assumed to support multi-thread programming at the user interface level. In particular, both the source data and the target data are shared and accessed by multiple threads.

Two key components in the parallel NUFFT are data partition and parallel scheduling. In data partition, we explore the potential concurrency in computation. For numerical correctness, data partition and parallel scheduling must support mutual exclusion in data updates. Initially, we partition the frequency space within the sphere into octants, with partition boundaries on the axial planes. This partition applies to both S and $\bar{S}$. A modification to this octal partition is necessary to avoid faults in concurrent modifications to the same datum. Specifically, a single source sample at s_j effects up to w_c translated samples on the Cartesian grid.

$$\begin{aligned} \mathbf{u}(\bar{s}_k) &= \mathbf{u}(\bar{s}_k) + \mathbf{C}(\bar{s}_k, s_j)\mathbf{u}(s_j), \\ &\quad \|\bar{s}_k - s_j\|_\infty < w_c/2. \end{aligned} \tag{3}$$

A Cartesian grid point on or close to the axial planes gets updated by its source neighbor points in more than one octants. Its datum value is therefore subject to potential errors in concurrent updating.

We establish an exclusion buffer between the octants. The source samples in each octant are either in an interior portion or in the buffer. Any sample in the interior portions is at least $w_c/2$ away from any other octant. We then impose a temporal or scheduling condition that the translation of the samples in an octant interior portion be not concurrent with that in the buffer. For instance, the translation of the interior samples may be carried out by up to eight threads, without any overlap in their accesses to the Cartesian grid points. This may be followed or preceded by the translation of the buffer samples. This simple scheme is implemented on a G5 quad-core processor at 2.5 GHz with 12 GB RAM and an AMD Opteron 8-core processor at 3.0 GHz with 8GB RAM. There are 421 million samples in the radial data, from a medical imaging application. The execution time on the 8-core processor is reduced by 7.33 times.

Consider next on multithreading the translation of the buffer samples as well. This entails a further partition in source data and a scheduling rule for mutual exclusion in memory accesses and data updates. We categorize the buffer samples according to their degrees in octant interaction. A buffer sample is said to have interaction degree 2 if it is between two and only two interior octants. Geometrically, the source samples of degree 2 occupy 12 plate zones, each axial plane is associated with 4 of the plates. The source samples of degree 4 fill the pencil gaps between any two degree-2 buffer zones, with 2 pencils along each axis. Finally, the remaining buffer samples are at or around the origin, with degree 8 in octant interaction. In retrospect, the degree of the interior samples is 1.

In short summary, the three-dimensional radial samples can be spatially partitioned into 8 octant interior portions (degree-1 zones), 12 plates (degree-2 zones), 6 pencils (degree-4 zones) and 1 center zone (degree-8). In parallel scheduling, the mutual exclusion condition can be satisfied by the unified rule : no permission of concurrent translations from source samples in zones of different interaction degrees. Alternative parallelization strategies will be discussed at the presentation.

References

[1] A. Dutt and V. Rokhlin. Fast fourier transforms for nonequispaced data. *SIAM Journal on Scientific Computing*, 14(6):1368–1393, 1993.

[2] G. Beylkin. On the fast fourier transform of functions with singularities. *Applied and Computational Harmonic Analysis*, 2(4):363–381, 1995.