Parallel Algorithms for 3D Reconstruction of Asymmetric Objects from Electron Micrographs

Robert E. Lynch, Dan C. Marinescu, Hong Lin
(rel, dcm, linh@cs.purdue.edu) Department of Computer Sciences, Purdue University
and
Timothy S. Baker
(tsb@bio.purdue.edu) Department of Biological Sciences, Purdue University

Abstract

We present new parallel algorithms for 3D reconstruction of objects from 2D projections and their application for the determination of the structure of macromolecules from electron micrographs. A multi-resolution orientation determination algorithm uses a parallel search to determine the 'best fit' of a given image with images in a reference database. The 3D reconstruction algorithm uses Cartesian coordinates and permits the reconstruction of objects that do not posses symmetries. The method decomposes a large linear system into a set of smaller systems that can be solved independently on different processors of a parallel computer or on a cluster of workstations. The paper outlines the reconstruction process and the data partitioning and load balancing issues pertinent to the parallel implementation of the algorithm; preliminary results obtained on a SGI Origin 2000 system are given.

Key words: Parallel processing, Structural biology, Electron microscopy, 3D reconstruction, Scientific computing.

1. Introduction. There are several practical methods to reconstruct a 3D object from a set of its 2D projections. We use a Fourier transform method [Cro70] implemented for parallel processing. Other methods include 'back projection' [Cro70] and numerical inversion of the Radon transform [She80]; see [Gor74] for a review of these and other methods. For descriptions of (sequential) methods for 3D reconstruction and many necessary related tasks, see [Dea93], [Fra96], and [Gra96] three of several books containing clear explanations and many references.

We are concerned with the 3D reconstruction of macromolecules, such as viruses. The experimental observations come from cryo-electron micrographs containing projections of many identical particles. These projections allow us to reconstruct the 3D electron density of the virus. Essential tasks include the preparation of the specimens, isolation of particular projections, the determination of the orientation of each projection, and so on. After an iterative process that involves improvement of the orientation and the reconstruction, an atomic model must be constructed which fits the reconstructed electron density.

3D reconstruction is a computationally and data intensive problem. Fourier coefficients in a sphere of an R-by-R-by-R uniform grid are determined by solving $O(R^2)$ linear least square problems each having $O(R)$ unknowns ("$O(R)$ denotes "order of magnitude of R"). R can be as large as 500 and routinely has values about 100. Each system can be solved independently and thus systems can be distributed among nodes. This is one of the justifications for designing parallel methods for 3D reconstruction.

A major challenge in the design of a parallel algorithm is finding a decomposition of a large problem into P smaller problems that can be solved as independently as possible on the P processors of a parallel system.

2. Macromolecule Reconstruction in Cryo-Electron Microscopy. The atomic structure determination of macromolecules based upon electron microscopy is an important application

Authorized licensed use limited to: Univ of Calif San Diego. Downloaded on March 08,2010 at 16:11:51 EST from IEEE Xplore. Restrictions apply.

of 3D reconstruction. X-ray crystallography and Nuclear Magnetic Resonance are other methods currently used for gathering experimental information about the 3D atomic structure of biological macromolecules. Traditionally, electron microscopy produced low resolution maps to, say, 20–30Å; but more recently, researchers at Cambridge and NIH were able to produce maps to 7–7.5Å resolution [Böt97, Con97]. There is hope that this resolution can be increased by using more projections for reconstruction, from the current level of a few hundred projections to thousands or tens of thousands.

Preparation of the specimen can take as few as several hours to complete for ideal samples, or, more typically, days or weeks for difficult specimens. Digitization can be performed at the microscope by recording images on a slow-scan CCD camera or by scanning images recorded on photographic film with a microdensitometer.

After digitized images are available, the procedure for structure determination is:

Step 1 Extract individual particle images from electron micrographs or CCD images.
Step 2 Determine the location of the projection of each particle center.
Step 3 Determine the orientation of each particle.
Step 4 Compute the 2D discrete Fourier transform of each projection.
Step 5 Compute the 3D discrete Fourier transform of the electron density (solve linear systems).
Step 6 Compute the inverse 3D transform to get the electron density.
Step 7 Analyze results; accept or refine by returning to Step 2.
Step 8 Dock an atomic model into the 3D density map.

The development of parallel algorithms to carry out some of these steps is part of an ambitious effort to design an environment for 'real-time electron microscopy', where results can be obtained in hours or days rather than in weeks or months.

Algorithms for Step 1, which include automatic identification of particle projections, are discussed elsewhere [Mar97]. Once particle projections are identified and extracted from micrographs (Step 1), the center and orientation of each particle projection are determined (Steps 2–3). The 2D discrete Fourier transform of each projection is computed (Step 4). With the orientation information, the points of intersection of the plane of projection and 3D grid lines can be determined and estimates of the 3D Fourier coefficients of the density at non-grid points are obtained by solving linear least squares systems (Step 5). An inverse 3D discrete Fourier transform is carried out to obtain estimates of the density at grid points (Step 6).

In Step 7, an analysis of the accuracy is carried out and the results are either accepted or a refinement is carried out. The map obtained in Step 6 can be improved by various refinement procedures and by the inclusion of more particle images and higher resolution information, if available. For example, an intermediate 3D reconstruction can be used as a model to obtain a more accurate center and orientation for each particle projection. This is then followed by Steps 4, 5, and 6, and additional cycles, as needed, to extend the resolution.

The time increases approximately linearly with the number of cycles of refinement (Steps 2-6). Typically, four or less cycles are sufficient for processing 'good' data at 20Å resolution. Model building, the process of fitting atomic level models into electron density maps, is achieved using atomic level information gathered from protein data banks (Step 8).

Much of the computation can be done independently from other parts. Each processor can be assigned a set of projections and carry out Steps 2–4 independently. Data exchange among nodes is necessary to collect information to do Step 5 and then each node solves its own set of linear systems. After doing a 2D inverse transformation of the solutions of the linear systems, another exchange is necessary before the final 1D inverse transformation to complete Step 5.

To use efficiently a parallel computer or a cluster of workstations, we need parallel algorithms which partition the computation uniformly, which minimize the communication among processors, and at the same time maintain a high level of locality of reference. This must be done not only for

Authorized licensed use limited to: Univ of Calif San Diego. Downloaded on March 08,2010 at 16:11:51 EST from IEEE Xplore. Restrictions apply.

the 3D reconstruction, but also for the determination of the orientation. Similar efforts have been reported in the past [Joh94], but the performance data available to us suggests that new algorithms have to be designed to reduce dramatically the computation time.

3. Parallel 3D Reconstruction. Our method for 3D reconstruction is modeled on one of the five methods suggested in [Cro70]; a preliminary version of our method is given in [Lyn97].

The observational data consist of a collection of digitized projections ('pixel frames') of a particle. Each projection is the integral of the electron density ρ along lines normal to the micrograph. We compute the 2D discrete Fourier transform of each pixel frame. This gives values, $P(u,v)$, at grid points in 2D Fourier (or 'reciprocal') space. These are the values of the 3D Fourier transform of the electron density on the plane of projection (see [Cro70] or any of the other references listed in §1). A grid point (u,v) in the the 2D reciprocal space is a point (h',k',ℓ') in the 3D reciprocal space. This leads to the linear system

$$P(u,v) = \sum_h \sum_k \sum_\ell F(h,k,\ell) \frac{\sin\pi(h'-h)}{\pi(h'-h)} \frac{\sin\pi(k'-k)}{\pi(k'-k)} \frac{\sin\pi(\ell'-\ell)}{\pi(\ell'-\ell)} \tag{3.1}$$

where $P(u,v)$ is a value of the discrete Fourier transform of the projection and $F(h,k,\ell)$ denotes a Fourier coefficient of the electron density of the object. One can use Fourier coefficients instead of a Fourier transform because the electron density is nonzero only in a sphere. The infinite sums are truncated and only values of h, k, ℓ in a ('resolution') sphere of radius R are kept. Since the density is real, the complex Fourier coefficients are conjugate symmetric, $\mathrm{conj}\{F(h,k,\ell)\} = F(-h,-k,-\ell)$, so F is needed only for nonnegative values of h, k, ℓ with $h^2+k^2+\ell^2 \le R^2$. But, because F has a real and an imaginary part, the number of real unknowns is about $4\pi R^3/3$. To obtain a solution of such a system by a direct method (i.e., a non-iterative method) typically requires $O(R^9)$ arithmetic operations (number of equations times the square of the number of unknowns).

[Cro70] point out that one could use values $P(u,v)$ at grid points to interpolate and obtain an estimate of P at a non-grid point (u',v') on the projection plane which corresponds to a point having (integer) grid coordinates k and ℓ in 3D. Then the equation above reduces to

$$P(u',v') = \sum_h F(h,k,\ell) \frac{\sin\pi(h'-h)}{\pi(h'-h)} \tag{3.2}$$

Typically $O(R^3)$ operations are needed to solve each system. Because there are about πR^2 systems (one for each pair (k,ℓ)), there are $O(R^5)$ operations to compute all of the F's. This is significantly less than the work used to solve (3.1), but here one has additional error due to the use of interpolation. To help reduce this error, the experimental data can be used three times, once to get (3.2) and once to get each of

$$P(u'',v'') = \sum_k F(h,k,\ell) \frac{\sin\pi(k'-k)}{\pi(\ell'-\ell)}, \qquad P(u''',v''') = \sum_\ell F(h,k,\ell) \frac{\sin\pi(\ell'-\ell)}{\pi(\ell'-\ell)} \tag{3.3}$$

In the first equation in (3.3), (u'',v'') on the plane is the same as the point (h,k',ℓ) in space, with h and ℓ integers; in the second, (u''',v''') and (h,k,ℓ') are the same point, with h and k integers.

A point (u,v) on a projection is a point (u,v,w) in 3D with $w=0$; the w-axis is normal to the projection. Knowledge of the center and the orientation of a projection (§2, Steps 2 and 3) leads to an invertible transformation matrix T which maps (h,k,ℓ) to (u,v,w). This is used to determine h' and corresponding point $(u'v')$; that is the linear relationship $T(h',k,\ell)^T = (u',v',0)^T$ is solved for h', u', and v' ("T" denotes transpose).

We summarize the calculation for constructing and solving (3.2) where k and ℓ are given integers. The bilinear interpolant of the transform of the pixel values is used to obtain the estimate $P(u',v')$. For each h', k, ℓ, inside the resolution sphere, the real and the imaginary parts of the complex

Authorized licensed use limited to: Univ of Calif San Diego. Downloaded on March 08,2010 at 16:11:51 EST from IEEE Xplore. Restrictions apply.

number $P(u', v')$ are stored together with h'. When this is done for all projections that intersect the grid line (k, ℓ), the linear system is (3.2) is formed. The (real) singular value decomposition least squares solver SGELSS from LAPACK [And92] is used to solve the resulting system for the real and the imaginary parts of $F(h, k, \ell)$, $h = 0, \pm 1, \ldots$.

4. Implementation of the Parallel Algorithms, Communication, and Load Balance. The implementation of both algorithms (Steps 2–3 and Steps 4–6) is based upon the 'Same Program Multiple Data' paradigm (SPMD). The critical issues for the performance of the parallel implementation of the algorithms are data partitioning, load balancing, and hiding the communication latency. Both programs consist of several computation phases. At the end of some of the phases, a global exchange of data occurs. To minimize synchronization delays the load should be balanced among processors for every execution phase.

We outline the implementation of the algorithms that determine the orientation and that carry out the 3D-reconstruction respectively. Orientation determination consists of construction of database and the database search. For any given iteration we first construct a low-resolution database and determine the 'best fit', during the global step of the algorithm. With hints from this step we construct a high-resolution database and conduct the search for each projection within a much smaller region of the entire database in the refinement step. For the global search the database is rather small and it is constructed independently by each node. In the refinement step the database is much larger and each node builds a section of it, concurrently with other nodes. In both cases each node is assigned a region of projections and determines the 'best fit' for each projection.

We have designed the algorithm for 3D reconstruction to work well on Origin 2000 (see §5, Table 2). The program uses MPI (Message-Passing Interface) and consists of several phases including initialization, 2D Fourier analysis, forming the systems of linear equations, solving linear systems, and Fourier synthesis. In the second phase of the algorithm, pixel frames are distributed evenly among nodes and processed independently in each node. The interpolants used to form the linear equations are calculated and collected for all three directions in a node. These data need to be put together to form a linear system of equations, such as (3.2). The systems are distributed among nodes so that after the systems are solved, a 2D Fourier synthesis can be done without an exchange. Then data must then be exchanged so that a 1D synthesis can be done to obtain the estimate of the electron density.

5. Experimental Results. Table 1 lists the major phases of the global search step of the orientation determination algorithm together with the time spent during each phase for one particular run. The measurements indicate that the database search dominates the execution time. Fortunately, this phase can be carried out in parallel therefore we expect decent speedups for large databases.

Phases		**Execution time (Secs)**
(1) Initialization	(seq, I/O)	1.52
(2) Construct the database	(par)	7.52
(3) Data exchange	(com)	2.98
(4) Database search	(par)	26.42
(5) Collect/write results	(com, I/O)	1.80

Table 1. The execution time in seconds for each phase of the global search in orientation determination. (seq) indicates a sequential execution phase of the program, (par) a parallel execution phase, and (com) a communication phase, (I/O) input-output.

Table 2 lists the phases of the 3D reconstruction program and preliminary measurements of the time spent in each phase during a particular run. The computation used 5000 projections, each having 51×51 pixels to reconstruct the density at points of a $51 \times 51 \times 51$ grid. The preliminary results reported here were obtained with data generated internally and there was no I/O in Phase

Authorized licensed use limited to: Univ of Calif San Diego. Downloaded on March 08,2010 at 16:11:51 EST from IEEE Xplore. Restrictions apply.

(2).

Phases		**Execution time** (Secs)
(1) Initialization	(seq, I/O, com)	0.09
(2) Read/Broadcast pixel frames	(I/O, com)	5.79
(3) 2D Fourier analysis	(par)	25.21
(4) Interpolation for setting linear systems	(par)	5.69
(5) Data exchange for solving linear system	(com)	7.43
(6) Solve linear systems	(par)	107.73
(7) 2D Fourier synthesis	(par)	0.60
(8) Data exchange for 1D Fourier synthesis	(com)	0.01
(9) 1D Fourier synthesis	(par)	0.31
(10) Gather and write density	(seq, I/O, com)	41.98

Table 2. The execution time in seconds for each phase of the 3D reconstruction reported by the master node. (seq) indicates a sequential execution phase of the program, (par) a parallel execution phase, and (com) a communication phase, I/O an Input/Output phase.

Preliminary results obtained on an Origin 2000 system indicate that our load balancing strategy for the 3D reconstruction discussed above works well.

Table 3 lists the execution time for two problems running on 8 nodes of an Origin 2000 system. Our results indicate that the algorithm is capable of quasi-linear speed up when the last phase for displaying/writing results is omitted. For both problems, we obtained nearly a six-fold speed up with eight nodes.

	Node 1	Node 2	Node 3	Node 4	Node 5	Node 6	Node 7	Node 8
A	44.1	41.0	41.0	41.0	41.0	41.0	41.0	41.0
B	195.0	153.3	153.3	153.3	153.3	153.3	153.3	153.3

Table 3. The execution time in seconds on each of the eight nodes of an Origin 2000 system for two problems. Problem A: 2,000 projections, each 21×21. Problem B: 5,000 projections, each 51×51.

6. Conclusions. The 3D reconstruction of asymmetric objects is a computationally and data intensive problem. It consists of iterative cycles of orientation determination and 3D reconstruction phases.

The algorithm for orientation determination consists of two steps, the global search and the refined search. For both steps a reference database is constructed and each projection is matched against all database components to get a best fit. For the global search a rather small database is constructed independently by every node and then each node is assigned an equal number of images. For the refined search both the database construction and the search are done in parallel.

The algorithm for 3D reconstruction is based upon an interpolation from the 2D projection Fourier domain to points on grid lines in the 3D density Fourier domain. To estimate values at grid points inside a sphere in the 3D density Fourier domain, we solve $O(R^2)$ linear least squares problems each having $O(R)$ unknowns. These problems are distributed among nodes and preliminary results indicate that the computational load is evenly distributed among nodes.

Profiling of the 3D reconstruction program indicates that algorithmic load balance is a necessary but not a sufficient condition for optimal execution. Though we have designed an algorithm that enables us to distribute the computational load evenly among nodes, practical considerations require I/O operations to be carried out by a single node when running the program on a cluster of workstations. The time for reading the input and writing the output data can be reduced significantly on a system supporting a parallel file system.

7. Acknowledgments The authors express their gratitude to Prof. Michael Rossmann for an

Authorized licensed use limited to: Univ of Calif San Diego. Downloaded on March 08,2010 at 16:11:51 EST from IEEE Xplore. Restrictions apply.

early version of his manuscript on 3D reconstruction [Ros98]. We are grateful for the contributions of K. C. van Zandt to a version of the parallel orientation algorithm. The work reported in this paper was partially supported by a grant from the National Science Foundation, MCB-9527131, by the Scalable I/O Initiative, by a grant from the Computational Science Alliance, and by a grant from the Intel Corporation.

References

[And92] Anderson, E., Z. Bai, C. Bischof, J. Demmel, J. Dongarra, J. Du Croz, A. Greenbaum, S. Hammarling, A, McKenney, S. Ostrouchov, and D. Sorenson, LAPACK Users' Guide, SIAM Publications, 1992.

[Böt97] Böttcher, B., S. A. Wynne, and R. A. Crowther, "Determination of the fold of the core protein of hepatitis B virus by electron cryomicroscopy", Nature (London) 386, 88–91.

[Con97] Conway, J. F., N. Cheng, A. Zlomick, P. T. Wingfield, S. J. Stahl, and A. C. Steven, "Visualization of a 4-helix bundle in the hepatitis B virus capsid by cryo-electron microscopy". Nature (London) 386, 91–94.

[Cro70] Crowther, R. A., D. J. DeRosier, and A. Klug, "The Reconstruction of a Three-Dimensional Structure from Projections and its Application to Electron Microscopy", Proc. Roy. Soc. Lond. A 317 (1970), 319–340.

[Dea93] Deans, S. R., "The Radon Transform and Some of Its Applications", 2nd Edit., Krieger Publishing Company, 1993.

[Fra96] Frank, J., "Three-Dimensional Electron Microscopy of Macromolecular Assemblies", Academic Press, 1996.

[Gor74] Gordon, R., "Three-Dimensional Reconstruction from Projections: A Review of Algorithms", Intern. Rev. of Cytology 38 (1974), 111–151.

[Gra96] Grangeat, P., and J-L Amans, Eds., "Three-Dimensional Image Reconstruction in Radiology and Nuclear Medicine", Kluwer Academic Publishers, 1996.

[Joh94] Johnson, C. A., N. I. Weisenfeld, B. L. Trus, J. F. Conway, R. L. Martino, and A. C. Steven, "Orientation Determination in the 3D Reconstruction of Icosahedral Viruses Using a Parallel Computer", IEEE (1994) 555–559.

[Lyn97] Lynch, R. E., and D. C. Marinescu, "Parallel 3D reconstruction of spherical virus particles from digitized images of entire electron micrographs using Cartesian coordinates and Fourier analysis", CSD-TR #97-042, Department of Computer Sciences, Purdue University, Aug. 1997.

[Mar97] Martin, I. M., D. C. Marinescu, T. S. Baker, and R. E. Lynch, "Identification of Spherical Particles in Digitized Images of Entire Micrographs", J. of Structural Biology, 120, 1997, 146–157.

[Ros98] Rossmann, M. G., and Y. Tao, "Cryo-Electron Microscopy Reconstruction of Partially Symmetric Objects", manuscript, May 29, 1998.

[She80] Shepp, L. A., "Computerized Tomography and Nuclear Magnetic Resonance" J. Comput. Assit. Tomog. 4 (1980), 94–107.

Authorized licensed use limited to: Univ of Calif San Diego. Downloaded on March 08,2010 at 16:11:51 EST from IEEE Xplore. Restrictions apply.