A Computational Framework for the 3D Structure Determination of Viruses with Unknown Symmetry.

Dan C. Marinescu and Yongchang Ji
School of Electrical and Computer Engineering
University of Central Florida
Orlando, Florida, 32816
Email: {dcm,yji}@cs.ucf.edu

April 2, 2003

Abstract

The protein shell of all virus structures resolved to this date exhibit some form of symmetry, most of them are spherical viruses with icosahedral symmetry; the study of viruses whose symmetry is not known, of structures which do not exhibit any symmetry, as well as the study of the genetic material of a virus are considerably more challenging. Increasing the resolution of the structure determination and solving structures with no symmetry represents a quantum leap in virus structure determination based upon electron microscopy. Computing is a major component of the structure determination process. Nowadays it is not feasible to increase the resolution of the structure determination of large macromolecules like viruses or to solve structures with no symmetry without novel parallel algorithms and environments enabling structural biologists to use parallel systems, clusters of workstations, or providing access to grid computing.

1 Introduction

The functional properties of a protein depends upon its 3D structure. To understand the biological function of a protein we need either to predict its 3D dimensional structure knowing the amino-acid sequence, or to gather experimental evidence and reconstruct the 3D structure of the protein. The *folding problem* is yet to be solved and the only alternative left to the structural biology community is to assiduously collect experimental evidence regarding the 3D structures of proteins.

Computing is a major component of the structure determination process. Nowadays it is not feasible to increase the resolution of the structure determination of large macromolecules like viruses or to solve structures with no symmetry without novel parallel algorithms and environments enabling structural biologists to use parallel systems, clusters of workstations, or providing access to grid computing.

It is instructive to observe the landscape of the collaborative research in biology and computer science. Genomics and protein sequence analysis have captured the interest and imagination of a fair number of computer scientists. Computational structural biology is an area less appealing to computer science probably because it requires a deeper understanding of the biological phenomena and of the experimental techniques used to capture empirical evidence regarding virus structures. At the same time, there are only a few laboratories in the world investigating the structures of viruses.

Our work is the result of a collaboration started in late 1980's with the Structural Biology group at Purdue University. The contribution of this paper is a computational framework for the 3D structure determination of large macromolecular structures like viruses with unknown symmetry using experimental evidence collected by cryo-transmission electron microscopy (cryo-TEM), to our knowledge the first framework of this kind.

The paper is organized as follows. Section 2 outlines basic concepts related to protein and virus structure. Section 3 provides the motivation for the study of asymmetric structures and discusses related work. Section 4 introduces the reader to the basic techniques for structure determination of large macromolecules like viruses and outlines the need for parallel algorithms. In Section 5 we describe three parallel algorithms for 3D reconstruction and present a succinct analysis of their computational and communication complexity as well as their space requirements. A parallel orientation refinement algorithm is presented in Section 6 and the agent based coordination environment is outlined in Section 7. Finally, we present our conclusions and discuss future work.

2 Protein and Virus Structure and Symmetry

The amino acid sequence of the peptide chain is called the *primary structure* of a protein. Regions of the primary structure form *secondary structures* such as α-helices and β-sheets. In turn, secondary structures are grouped together into globular domains known as *tertiary structures*; a protein may also contain several polypeptide chains arranged in a *quaternary structure*. Amino acids far apart in the linear structure are brought together in tertiary and quaternary structures forming *functional regions* [Bra91].

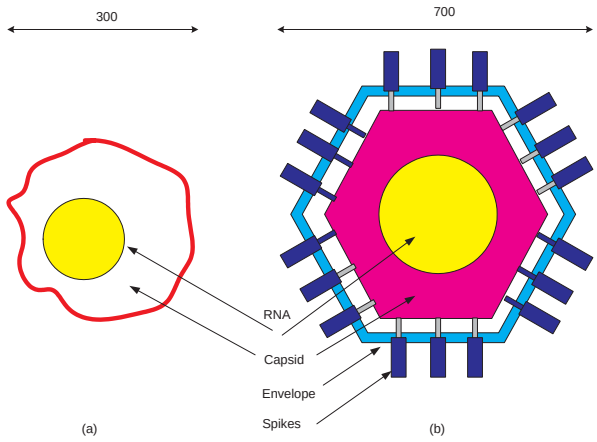

Figure 1: Viruses differ in terms of genome, molecular mass, relative weights of the protein shell and the genome, shape, and size as shown by the schematic diagrams of two spherical viruses. (a) A *picona* virus with a diameter of about 300 Å. (b) A *Sindbis* virus with a diameter of about 700 Å.

The primary structure of a protein can be inferred from the DNA sequence. Nowadays, recombinat DNA techniques allow fast determination of the DNA sequence from structural genes. Thus, the number of proteins whose primary structure is known grows exponentially.

The functional properties of a protein are determined by its three-dimensional structure. The *active sites* formed when secondary structures are joined to form tertiary and quaternary structures

are critical in performing various functions. For example, the *binding site* of a virus determines which cells can be infected.

Unfortunately, the *folding problem*, the ability to predict the 3D structure of a protein from the sequence of amino acids, is yet to be solved. Similar structural domains can be created by different amino acid sequences; at the same time, slight variations of the primary structures fold differently.

A preeminent introductory text in structural biology [Bra91] describes the relationship between the primary and the 3D structure as follows: "these sequences tell little more about the biology of the system than a New York City telephone directory tells about the functions and marvels of that city".

The structure of the DNA is relatively simple and predictable, it consists of only four different nucleotides. On the other hand, the 20 amino acids, the building blocks of a protein lead to a very large number of ways in which similar domains can be generated. As pointed out earlier, the 3D structure of proteins cannot be predicted theoretically, it can only be determined experimentally.

Viruses are large macromolecules that cause a variety of human, animal, and plant diseases. A virus consists of a nucleic acid genome, a protein shell or *capsid*, and sometimes a membrane or *envelope*, which can enclose or be enclosed by the capsid, see Figure 1.

The protein shell of spherical viruses has icosahedral symmetry. An icosahedron, see Figure 2 (a), has 12 vertices and an equal number of 5−fold axes, 20 faces and the same number of 3-fold axes; it has 30 edges and the same number of 2−fold axes. An object placed on an icosahedron is repeated 60 times by combining the symmetry operations of the 5−fold, 3−fold and 2−fold axes. Indeed the capsid consists of $T \times 60$ sub-units where T can be 1 or more. Caspar and Klug have shown that the *triangulations* number $T = h^2 + hk + k^2$, with h and k integers, [Bra91]. Possible values for T are $1, 3, 4, 7, 9, 12, 13, \ldots$.

For $h = k = 1$, $T = 3$ and the protein shell consists of $3 \times 60 = 180$ sub-units. Each asymmetric unit consists of three proteins A,B, and C, see Figure 2 (b). In this case the RNA contains the genetic blueprint of the three proteins, A, B, and C.

A virus has to locate a specific docking site on a host cell. The information about the structure of the capsid is critical in determining the pharmaceutical compounds that can block the virus binding site. Thus, the structure of viruses is not only of scientific interest but it has the potential to lead to the discovery of antivirals for the prevention and treatment of plant, animal, and human diseases.

3 Motivation and Related Work

The protein shell of spherical viruses exhibit various degrees of symmetry, but the core of the virus, consisting of genetic material, does not. Structural studies of the virus core require 3D reconstruction algorithms of asymmetric objects. However, the study of asymmetric objects requires a considerable increase in the amount of experimental evidence and in the complexity of the numerical methods to determine the structure; thus the need for parallel algorithms for structure determination.

On the other hand, there is a need to increase the resolution of structure determination based upon electron microscopy. X-ray crystallography has been the only method to obtain high resolution electron density maps, at 3 Å or better, for large macromolecules like viruses. Until recently electron microscopy was only able to provide low resolution maps, 20 Å or lower. cryo-TEM is appealing to structural biologists because crystallizing a virus is sometimes impossible and even when possible, it is technically more difficult than preparing samples for microscopy. Thus the

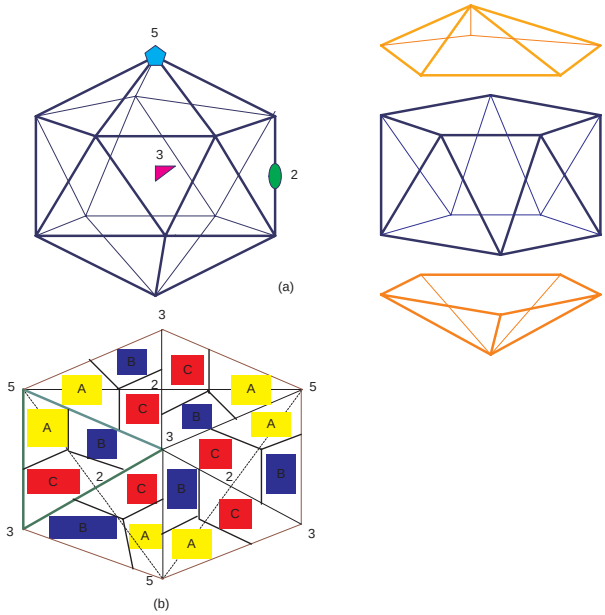

Figure 2: The protein shell of spherical viruses have icosahedral symmetry. (a) An icosahedron has twelve, (12), 5−fold axes of symmetry one through each vertex, twenty, (20), 3−fold axes, each going through the center of each face, and thirty, (30), 2−fold axes, each going through the half point of an edge. (b) For $T = 3$ each asymmetric unit (shown with tick lines) consists of three proteins A,B, and C

desire to increase the resolution of cryo-TEM methods to the 5 Å range. Recently, successful results in the 7–7.5 Å range have been reported, [Böt97], [Con97]. But increasing the resolution of the 3D reconstruction process requires more experimental data. It is estimated that the number of views used to obtain high resolution electron density maps from cryo-TEM micrographs should increase by two order of magnitude from the current hundreds, to tens of thousands.

The amount of experimental data further increases when structural studies of even larger virus-antibody complexes are attempted. The use of larger zero fill ratios to improve the accuracy of 3D reconstruction [MaY01, MaY02] increases the number of arithmetic operations and the amount of space needed for reconstruction.

While a typical reconstruction of the shell of an icosahedral virus at, say, 20 Å resolution may require a few hundreds views, e.g., 300, the reconstruction of an asymmetric object of the same size and at the same resolution would require 60 times more data, or 18,000 views. See [Ros98] for the minimal number of views as a function of the desired resolution and the size and symmetry of the virus.

Thus it is not unrealistic to expect an increase in the volume of experimental data for high resolution asymmetric objects by three to four orders of magnitude in the near future. Even though nowadays faster processors and larger amounts of primary and secondary storage are available at a relatively low cost, the 3D reconstruction of asymmetric objects at high resolution requires computing resources, CPU cycles, and primary and secondary storage, well beyond those provided by a single system. Thus good parallel algorithms are needed.

The Radon transform [Rad17] provides an analytical solution to the problem of reconstructing a 3D object from a set of 2D views. In late 1960's and early 1970's the interest in 3D reconstruction

was renewed by the introduction of computerized tomography in medicine and by the electron microscopy [DeR67]. The group of Aaron Klug at Cambridge presents analytical results and makes use of the "projection or central section theorem", [KaS88] and [Hhb81], as well as the experimental reconstruction of helical fibers [DeR70], and [Cro70] gives five methods, together with the simplification resulting when symmetry is exploited by the use of Bessel functions e.g., for icosahedral viruses.

There are several practical methods for reconstructing a 3D object from a set of its 2D views. The methods for 3D reconstruction include use of Fourier Transforms, back projection, and numerical inversion of the Radon Transform. See [Gor74] for a review of these and other methods. Also, for descriptions of (sequential) methods for 3D reconstruction and related tasks, see [Dea93], [Fra96] and [Ram71], three of several books containing clear explanations and many references.

There are subtle differences between the applications of 3D reconstruction in computerized tomography and in cryo-TEM for virus structure determination. There are two basic approaches used in electron microscopy: single-particle and multiple-particle techniques. The multiple-particle approach used in cryo-TEM requires methods to determine the orientations of individual particles as discussed in Sections 4 and 6

The classical tomographic approach requires many views of the same particle tilted in various orientations. In case of computer assisted tomography, a computer controls the angles and the entire data collection process. The tilts required by these approaches limit the resolution level. The single-particle approach is described in [Heel00].

The filtered back projection algorithm was proposed by Bracewell and Riddle [BrR67], and later independently with additional insights by Ramanchandran and Lakshminarayanan in [Ram71] who are also responsible for the convolution back-projection algorithm. This algorithm is now used by almost all commercially available CT scanners [KaS88]. The filtered back projection algorithm [Ram71] uses Fourier theory to arrive at a closed form solution to the problem of finding the linear attenuation coefficient at various points in the cross-section of an object.

The basic ideas of the maximum likelihood methods for 3D reconstruction is that the best model is the most consistent with the observations. The consistency is measured by the probability that the observation would be made given the current model. When the model is modified to make the observations more probable, the model gets better and the likelihood increases. The probability that the current model leads to the actual observations includes the effects of all sources of errors and as the model improves the errors decrease. Maximum likelihood methods for protein structure determination are discussed in [Pan96], [Mur97] and [Bri96].

A recent paper covers the subject of ab initio reconstruction for cryo-TEM [Doe00]. The paper introduces a statistical model and uses maximum likelihood reconstruction. The 3D structure of several viruses is computed using an expectation maximization algorithm.

To use efficiently a parallel computer or a cluster of workstations, we need parallel algorithms that partition the data and computations evenly among nodes to ensure load balance and, moreover, which minimize the communication among processors by maintaining a high level of locality of reference. Similar efforts have been reported in the past [Joh94] and [Ram95], but the performance data available to us suggest that new algorithms have to be designed to reduce dramatically the computation time, communication time and to support reconstruction of asymmetric particles.

4 Structure Determination in Cryo-TEM

Electron microscopy is a widely used method for obtaining experimental evidence regarding the structure of viruses at low to moderate resolution [Bak99], [Cro70], [DeR67], [DeR70], [Fra96], [Heel00], [Ram71]. In cryo-transmission electron microscopy (cryo-TEM) a solution containing virus particles is vitrified at liquid nitrogen temperature and the sample is placed inside the microscope where it is irradiated with an electron beam that forms an image on film or on a CCD. An entire micrograph consists of real images of many identical virus particles frozen in the sample in different orientations [MaM97].

4.1 The 3D Reconstruction Procedure

Reconstructing the 3D image of the virus from the 2D views is conceptually similar with Computed-Aided Tomography (CAT) [Gor74], [Gra96], [Her79]. The notable difference is that the orientations of the 2D images are known in CAT (the patient is in a fixed position and the images are taken at known angles of the X-ray source) while in cryo-TEM virus particles are frozen in the solution at random orientations. Moreover, we assume that all virus particles frozen in the sample are *identical* thus we have many 2D views of a single object whose 3D electron density we wish to reconstruct. In CAT we have multiple 2D images of a single object.

The procedure for 3D structure determination in cryo-TEM consists of the following steps:

Step A Extract individual particle views from micrographs and identify the center of each view.

Step B Determine the orientation of each view.

Step C Carry out the 3D reconstruction of the electron density of the macromolecule.

Step D Dock an atomic model into the 3D electron density map.

Algorithms for Step A, which include automatic identification of particle views, the determination of the center and the orientation of each virus particle view are discussed elsewhere [MaM97]. In this paper we discuss only algorithms for Steps B and C.

Steps B and C are executed iteratively until the 3D electron density map cannot be further improved at a given resolution; then the resolution is increased gradually and the iterative execution of steps B and C is repeated. The number of iterations for 3D reconstruction is in the 10–20 range and one step of 3D reconstruction for a medium size virus may take several days. Typically it takes months to obtain a high resolution electron density map. Then Step D of the process takes place.

Best results in obtaining high resolution electron density maps are often obtained in the case of highly symmetrical particles such as icosahedral viruses because the high symmetry leads to redundancies in the Fourier transform data and that, in turn, aids the orientation search process.

It was estimated that approximately 2000 particle images are necessary for the reconstruction of a virus with a diameter of 1000 Å at 10 Å resolution [Ros98], and recent results at 7 - 9 Å resolution for Hepatitis B virus capsides [Böt97], [Con97] have confirmed this estimate.

4.2 The Quality of the Solution, the Contrast Transfer Function

A correction of the experimental data used for the 3D structure determinations is necessary. The relationship between the electron image of a specimen and the specimen itself is described by the *Contrast Transfer Function (CTF)* [Bak99]. The CTF that enables the visualization of unstained

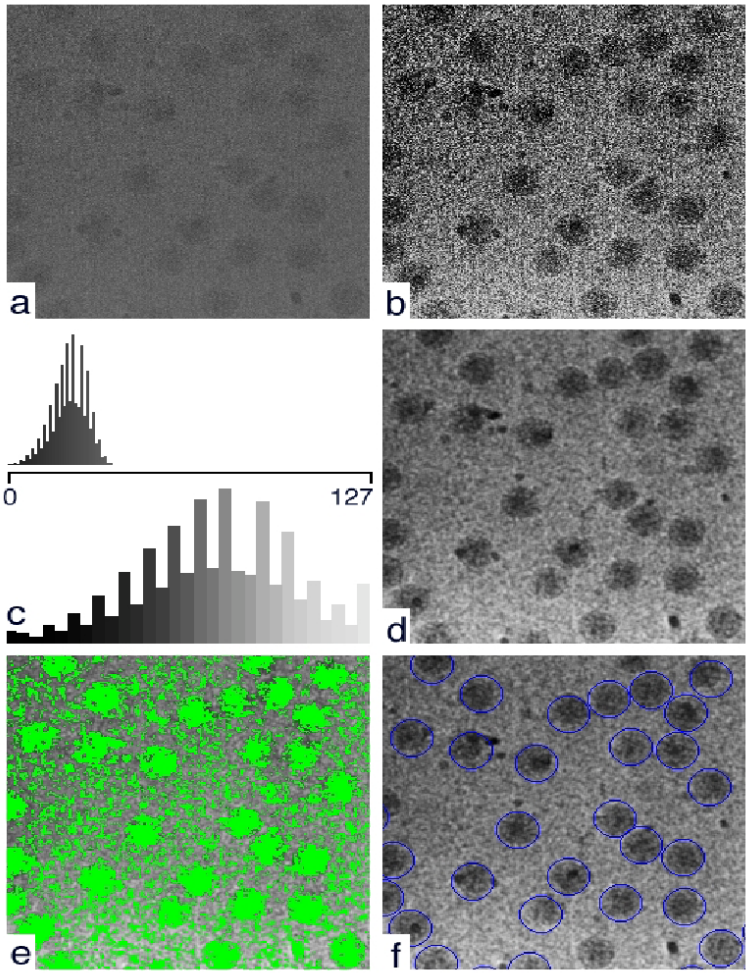

Figure 3: (a) Portion of low-contrast micrograph of frozen-hydrated sample of retrovirus cores. (b) The micrograph after histogram equalization. (c) Gray level histograms before (top) and after (bottom) histogram equalization. (d) The micrograph in (b) after neighborhood averaging with a 10×10 filter. (e) Contents of the binary image after pixel marking (green) superimposed on the micrograph in (d). (f) The result.

specimens must be compensated for in the reconstruction, in order to achieve a reliable representation of the structure. The transfer function reverses, removes, and attenuates data in the image. The effects of the transfer function become proportionally more significant at progressively higher resolutions. One can correct the transfer function by means of a variety of filtering methods [Gon93].

The CTF is a complex valued function. The phase and the amplitude are characteristic to a particular microscope, the specimen, and the conditions of imaging. The CTF includes both phase- and amplitude-contrast components [Bak99]:

$$CTF(v) = -(\sqrt{1 - F_{amp}^2} \times \sin(\chi(v)) + F_{amp} \times \cos(\chi(v)))$$

where v is the spatial frequency, (per angstrom), F_{amp} is the fraction amplitude contrast and $\chi(v))$ is the phase.

The CTF correction affects individual coefficients of the 2D Discrete Fourier Transform (DFT) of a view. Let $\mathcal{E}^i$ be a view and $\mathcal{F}^i = DFT(\mathcal{E}^i)$. Then $\mathcal{F}^i$ consists of set of Fourier coefficients: $\{f^i_{\mathbf{j},\mathbf{k}}\}, \quad 1 \leq (\mathbf{j},\mathbf{k}) \leq l$. The CTF-factor for the Discrete Fourier Transform (DFT) coefficient $f^i_{\mathbf{j},\mathbf{k}}$ is defined as:

$$\tau^i_{\mathbf{j},\mathbf{k}} = -\exp(c \times Tempfac \times (\mathbf{j}^2 + \mathbf{k}^2))/(\mid CTF \mid +\epsilon)$$

where c is a constant that depends on the pixel size and DFT size, $Tempfac$ gives an estimate of the temperature factor for the image data, and ϵ is a "fudge factor" that keeps the CTF correction from magnifying the noise in the data by a value greater that $1.0/\epsilon$.

The CTF-factor is a function of $Tempfac$ and r_{map} the resolution of the electron density map. We give more weight to higher frequency components at higher resolution (large radius in the Fourier domain) while computing the distance in orientation refinement program.

Our algorithms allow an effective CTF correction. In the interpolation phase we weight experimental data points coming from different micrographs with the corresponding value of CTF. Figure 4 shows cross-section 80-83 of $\phi X174$, a bacteriophage virus with icosahedral symmetry at 10 Å resolution without and with CTF correction. The improvement in the quality of the reconstruction is visible.

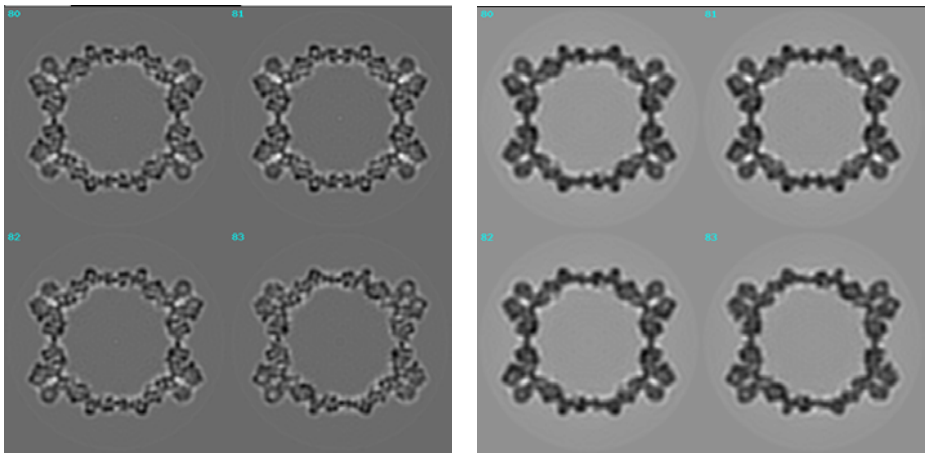

Figure 4: On the left: crosssections 80-83 of $\phi X174$, a bacteriophage virus with icosahedral symmetry, at 10 Å resolution and no CTF correction. On the right: the same crosssections with CTF correction.

5 3D Reconstruction

In this section we are only concerned with Step C of the process outlined above. We use a Fourier method based on the 'projection or central section theorem [Bak99]. We use the Discrete Fourier Transform (DFT) to approximate the Fourier Transform.

The process is carried out as follows:

Step I Compute the 2D DFT of each view.

Step II Use each 2D DFT and its estimated orientation to calculate by interpolation values of the 3D DFT at points within half a grid spacing from the plane of view. At each (h,k,ℓ), average the interpolated values obtained from views to obtain the final estimate of the 3D DFT.

Step III Compute the inverse 3D transform to get the electron density.

A 3D object has finite dimensions R and according to [Heel00], the 2D projection operation should be carried out only up to distance R from the center; the authors conclude that the central section in reciprocal space is no longer an infinitely "thin" plane but a "slab" with thickness $1/R$. Thus central sections overlap not only on a single line and the closer to the origin of the reciprocal space, the more the central sections used for reconstruction overlap. This implies that low frequency components receive too much weight in the reconstruction and the 3D volume will be "blurred". Thus we need a high-pass filtering of the views. van Heel warns that algorithms filling in missing information by maximum entropy or by POCS, Projections Onto Convex Sets, may lead to incorrect results, [Har86].

Our basic algorithm for the reconstruction of asymmetric objects in Cartesian coordinates, which is based on methods in [Cro70], is presented in [Lyn97] and [Lyn99]. An improvement of the algorithm that reduces the number of arithmetic operations by two orders of magnitude is outlined in [Lyn00], and a detailed presentation and analysis of the algorithms together with preliminary experimental results are given in [MaY01, MaY02]. In this, we consider the DFT of a pixel frame to be a slab and use bilinear interpolation to obtain values at points of the 3D grid.

The algorithm for 3D reconstruction was designed for clusters of workstations and for the SP2 parallel computer. The program uses MPI (Message-Passing Interface) and consists of several phases including initialization, 2D Fourier analysis, estimating the 3D coefficients of the DFT by interpolation, and Fourier synthesis.

5.1 Algorithms for 3D Reconstruction in Cartesian Coordinates

In this section, we present three algorithms for reconstruction which have different space, communication, I/O, and computational requirements [MaY01]. The three algorithms assume that the parallel system does not support parallel I/O operations. When a system with a parallel file system is used, then each processor can read directly that segment of the input data it needs, and, similarly, write that segment of the output data it produces. In the case of a system without a parallel file system this strategy leads to a high level of I/O contention.

The clusters of PCs and the SP2 system which we have been using do not have parallel file systems. To avoid the increased time which results from I/O contention, we use a Coordinator-Worker paradigm. The coordinator (a processor), C, reads the input views and their orientations and then distributes them to the workers, $W_i, 1 \leq i \leq P-1$; C uses gather operations to collect the final result.

The three algorithms outlined below use different strategies for distributing the input data to the workers and they use different data structures. These differences affect the amount of storage required and the execution time; experimental results are given in Section 5.2.

We use the following notations:

$(\mathbf{x}, \mathbf{y}, \mathbf{z})$: a grid point in 'real space'.

$(\mathbf{h}, \mathbf{k}, \mathbf{l})$: a grid point in 'reciprocal space'.

D : the current version of the electron density map of size l^3.

$\mathcal{D} = DFT(D)$: the 3D Discrete Fourier Transform (DFT) of the electron density map.

m : the number of experimental views.

$\mathcal{E} = \{\mathcal{E}^1, \mathcal{E}^2, \ldots \mathcal{E}^q, \ldots \mathcal{E}^m\}$: the set of experimental views.

$\mathcal{F} = \{\mathcal{F}^1, \mathcal{F}^2, \ldots \mathcal{F}^q, \ldots \mathcal{F}^m\}$: the set of 2D DFTs of experimental views. Here $\mathcal{F}^q = DFT(\mathcal{E}^q)$ for $1 \leq q \leq m$.

$l \times l$: the number of pixels in one view.

k, $k \geq 1$: zero fill aspect ratio; the $l \times l$ pixel array is put into a $(kl) \times (kl)$ array with zero fill before the DFT is computed.

P : the number of processors.

C : the coordinator, the processor performing the I/O operations.

W_i, $1 \leq i \leq P-1$: the processors that carry out tasks assigned to them by the Coordinator.

m : the number of views.

$v = \lceil m/P \rceil$: number of views processed by W_i in Algorithms I and III.

$v' = m - (P-1)v$: the number of views processed by C in Algorithms I and III.

q' : the number of views processed by one processor in Step 2 of Algorithm III.

$q = Pq'$: the number of views read in one I/O operation in Algorithm III.

$q_{stages} = m/q$: the number of pipeline stages in Algorithm III.

p : the edge length, in Å, of a pixel; grid spacing in real space.

$p' = 1/(kp)$: the grid spacing in reciprocal space.

r_{map} : the desired resolution in Å.

R : number of grid spaces in reciprocal space to obtain the desired resolution.

n : the order of the symmetry group; one view can be used n times with different orientations.

h-slab : a number of consecutive **kl** planes in $\mathcal{D}$ for $\mathbf{h}_{min} \leq \mathbf{h} \leq \mathbf{h}_{max}$. Similarly we define **k**-slabs and **l**-slabs.

Algorithm I. There are no replicated computations and there is no overlapping of communication with computations. Each processor must have space to hold the entire $\mathcal{D}$. The algorithm is space-intensive.

The algorithm consists of the followings steps:

1. C reads all m views, keeps v' views, and scatters a different group of v views to each processor W_i, $1 \leq i \leq P-1$.

2. C reads the orientation file and scatters a group of v orientations to the W_i, $1 \leq i \leq P-1$, so that each processor has the orientation of each view assigned to it.

3. After zero fill, each processor performs a 2D DFT for each of the v (or v') views assigned to it.

4. Each processor uses the entire 3D volume in the reciprocal space, and uses the 2D DFT of each view assigned to it to compute values at grid points scattered throughout the entire volume. It does so by intersecting each view with the **h**, **k**, and **l** lines in the entire reciprocal space. Then it estimates the 3D DFT of the electron density at grid points near the plane of each view and inside the hemisphere of radius R;

5. If the object has n-fold symmetry, then the interpolation of Step 4 is repeated for each of the other $n-1$ symmetrically related orientations of the 2D DFT of each view.

6. The P processors perform a global exchange. At the end, the i-th processor has its **h**-slab of interpolated values, as computed by it and by each of the other processors. It averages these values and obtains the final estimate of the `3D DFT` at all grid points in its **h**-slab and inside the hemisphere of radius R.

7. The i-th processor performs $k \times l/P$ inverse `2D DFT`s in its **h**-slab.

8. The P processors perform a global exchange and each ends up with its **h**-slab.

9. Each processor performs $\frac{(k \times l)^2}{P}$ `1D DFT`s to obtain the final estimate of the electron density in its z-slab.

10. Each W_i, $1 \leq i \leq P-1$, sends its slab of electron density to C.

11. C writes the electron density map D to a file.

The computation complexity for each processor by Algorithm I is:
$O(\ \lceil m/P \rceil\ (k\,l)^2 \log(k\,l)\ +\ n\ \lceil m/P \rceil\ R^2\ +\ klR^2\ +\ (k\,l)^3 \log(k\,l)/P\ )$
The space required for each processor is: $O(\ \lceil m/P \rceil\ (k\,l)^2\ +\ (k\,l)^3\ )$
Algorithm I performs two *scatter* operations at the beginning, followed by two *scatter-gather* operation and one *gather* operation at the end. The communications complexity of Algorithm I is:
$O(\ m\,l^2\ +\ (k\,l)^3\,P\ )$

Algorithm II. This algorithm attempts to minimize the amount of space. Each processor calculates the `2D DFT` of each view. The algorithm is computation intensive, computations are duplicated to reduce space and communication requirements. There is no overlapping of communication with computations.

1. C reads the orientation file.

2. C reads one view at a time and broadcasts it to W_i, $1 \leq i \leq P-1$.

3. C broadcasts the values of the orientation so that each W_i, $1 \leq i \leq P-1$, has the orientation of the view assigned to it.

4. After zero fill, each processor performs a `2D DFT` for the view.

5. Each processor maintains a copy of the **h**-slab and uses the `2D DFT` of each view to fill out its slab. It does so by intersecting each view with the **h**, **k**, and **l** lines in its **h**-slab and inside the hemisphere of radius R.

6. If the object exhibits n-fold symmetry, Step 5 is repeated for all the $n-1$ other views related by symmetry to the one obtained experimentally.

7. Steps 2 to 6 are repeated until all views have been processed. At the end of these steps, each processor has all the values inside the **h**-slab allocated to it.

8. The i-th processor performs kl/P inverse `2D DFT`s in its **h**-slab.

9. The P processors perform a global exchange and each ends up with a **k**-slab.

10. Each processor performs $(k\,l)^2/P$ `1D DFT`s to obtain the final estimate of the electron density in its **k** -slab.

11. Each W_i, $1 \leq i \leq P-1$, sends its slab of electron density to C.

12. C writes the electron density map D to a file.

The computation complexity of algorithm II is:
$O(\ m\,(k\,l)^2\,\log(k\,l)\ +\ n\,m\,R\,k\,l/P\ +\ (k\,l)^3\log(k\,l)/P\ )$
The space required for each processor is: $O(\ (k\,l)^2 + (k\,l)^3/P\ )$
Algorithm II performs two *broadcast* operations at the beginning, followed by one *scatter-gather* operation and one *gather* operation at the end. The communications complexity of Algorithm II is: $O(\ P\,m\,l^2 + (k\,l)^3\ )$

Algorithm III. This pipelined algorithm reduces the amount of space and allows overlapping of I/O and computations. No computations are duplicated.

1. C reads the orientation file.

2. C reads groups of q views at a time and scatters them in groups of size q' to W_i, $1 \leq i \leq P-1$.

3. C scatters the values of the orientations so that each W_i, $1 \leq i \leq P-1$, has the orientation of each view assigned to it.

4. After zero fill, each processor performs a `2D DFT` for each of the v views assigned to it.

5. Each processor uses the information provided by the subset of views to compute grid points scattered throughout the entire `3D` volume. By intersecting each view with the **h**, **k**, and **l** lines, it obtains all $\mathcal{D}(\mathbf{h},\mathbf{k},\mathbf{l})$ with **h**, **k**, **l** within half a grid spacing from the view, and stores the $\{\mathbf{h},\mathbf{k},\mathbf{l},\mathcal{D}(\mathbf{h},\mathbf{k},\mathbf{l})\}$ 4-tuples.

6. If the object exhibits n-fold symmetry, Step 5 is repeated, using the `2D DFT` of Step 4, and each of the $n-1$ symmetrically related orientations.

7. Each processor retains those $\{\mathbf{h},\mathbf{k},\mathbf{l},\mathcal{D}(\mathbf{h},\mathbf{k},\mathbf{l})\}$ 4-tuples within the **h**-slab allocated to it and exchanges each of the others with the corresponding processors, then each processor convert the 4-tuples value in to its own **h**-slab grid value inside the hemisphere of radius R.

8. Steps 2 to 7 are repeated until all input views are processed. At that time, at the end of Step 7, each processor has all the values of the 3D `DFT` at the grid points inside its **h**-slab.

9. The i-th processor performs kl/P inverse `2D DFT`s in its **h**-slab.

10. The P processors perform a global exchange and each ends up with its **h**-slab.

11. Each processor performs $(k\,l)^2/P$ `1D DFT`s to obtain the final estimate of the electron density in its k-slab.

12. Each W_i, $1 \leq i \leq P-1$, sends its slab of electron density to C.

13. C writes the electron density map D to a file.

The computation complexity of Algorithm III is:
$O(\ \lceil m/P \rceil\ (k\,l)^2\,\log(k\,l)\ +\ n\ \lceil m/P \rceil\ (R^2 + (k\,l)^2)\ +\ (k\,l)^3\log(k\,l)/P\ )$
The space required for each processor is: $O(\ n\,q'\,(k\,l)^2 + (k\,l)^3/P\ )$
Algorithm III performs two *scatter* operations at the beginning, followed by $q_{stages}+1$ *scatter-gather* operations and one *gather* operation at the end. Recall that q_{stages} is the number of pipeline stages. The communications complexity of Algorithm III is: $O(\ m\,(k\,l)^2\ +\ (k\,l)^3\ )$

5.2 Performance Data

Our goals were: (a) to check the quality of our results, and (b) to gather performance data and asses the relative merits of each of the three algorithms. Details regarding our measurements can be found in [MaY01, MaY02]. The experimental virus data are presented in Table 1; viruses exhibit either icosahedral or dihedral symmetry and the number of views ranges from 60 to slightly less than 2,000.

We present performance data collected on an IBM SP2 system with 64 nodes, each with four `POWER` III 375 Mhz processors. Each node has 4 GBytes of main storage and a 36.4 GB disk. The 4 processors in each node share the node's main memory and communicate, using `MPI`, with local processors and with processors in a different node.

To check the quality of the solution we compared our results with the ones obtained with a reconstruction program used for many years by the structural biology community [Bak99]. A qualitative evaluation was done visually by comparing the electron density maps. For a quantitative evaluation we calculated the relative error of our maps compared with the ones created with the widely used program. The results of the visual inspection and those of the quantitative evaluation indicated that our algorithms are working properly. Our reconstruction algorithm allows an efficient implementation of the CTF correction, see Section 4.2, and the results reflect this advantage. In Figure 5 we show a central cross section of the electron-density maps obtained with our parallel program and traditional sequential program.

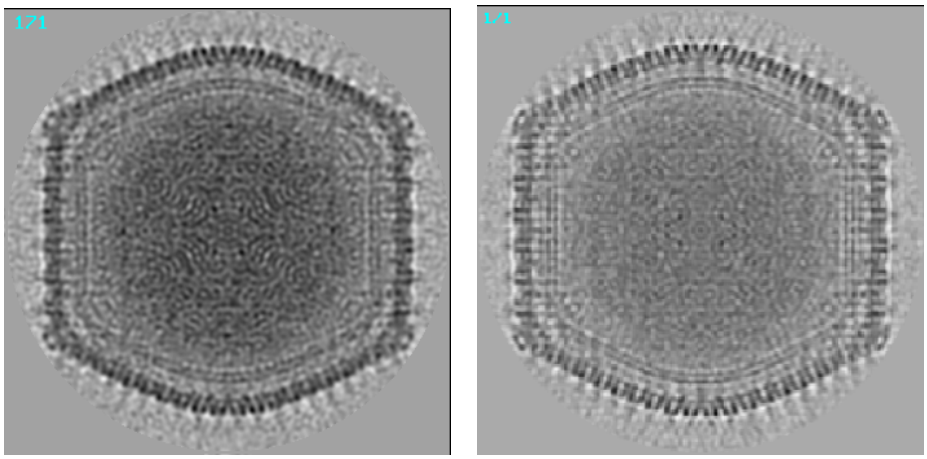

Figure 5: Crosssection 171 of CIV at 27 Å, obtained with our parallel program (left) and traditional sequential program(right) [Yan00].

To investigate the performance of our programs we measured the execution time and we recorded the memory usage for each of the three algorithms discussed in Section 5.1. Tables 2 and 3 show the execution time in seconds and the memory requirements in MB for the set of problems tested using Algorithms I and III respectively for a number of processors $P = 1, 2, 4, 8, 16, 32$. The execution times reported in Tables 2, and 3 do not include the time to write the `3D` electron density onto the output file. Figures 6 and 7 illustrate the speedups for the three algorithms. Table 4 presents ratio of the execution time and memory for Algorithms II and III versus Algorithm I.

The memory requirements of Algorithm I made it impossible to run Problems C, D and G with less than two processors and to run Problem E with less than 8 processors. The amount of memory varies only slightly when the number of nodes increases, the largest problem, Problem E, requires slightly less than 700 MB of memory per node while problem A requires less than 5 MB on each of

Table 1: Virus structures used to test the parallel `3D` reconstruction program on an IBM SP2. The virus type, the number of pixels, the number of experimental views and the types of symmetry are indicated. Note that for a virus with icosahedral symmetry there are 60 views related to a single experimental one; for a virus with dihedral symmetry from each experimental there are 10 related ones.

Problem	Virus (Virus Family)	Symmetry	Pixels	Views
A	Polyomavirus (Papovaviruses)	Icosahedral	99×99	60
B	Paramecium Bursaria Chlorella type 1	Icosahedral	281×281	107
C	Auravirus (Alphaviruses)	Icosahedral	331×331	1940
D	CIV Chilo Iridiscent	Icosahedral	343×343	667
E	Herpes	Icosahedral	511×511	173
F	Ross River Virus (Alphaviruses)	Dihedral	131×131	1777
G	Paramecium Bursaria Chlorella type 1	Icosahedral	359×359	948
H	Bacteriophage Phi29	Dihedral	191×191	609
I	Sindbis (Alphaviruses)	Icosahedral	221×221	389
J	Sindbis (Alphaviruses)	Icosahedral	221×221	643
K	Polyomavirus (Papovaviruses)	Icosahedral	69×69	158

Table 2: Algorithm I. Execution time (seconds) and memory requirements (MB) for P=1, 2, 4, 8, 16, and 32 processors.

P	Time/ Memory	A	B	C	D	E	F	G	H	I	J	K
1	Time	16.4	1126	–	–	–	437	–	471	519	813	28.0
	Mem	11.6	266	–	–	–	27	–	84	129	129	3.9
2	Time	8.4	566	3554	973	–	223	2335	241	263	413	14.1
	Mem	5.8	134	218	242	–	27	278	84	65	65	2.0
4	Time	4.5	288	1801	499	–	116	1186	126	136	210	7.4
	Mem	4.9	111	181	202	–	28	231	84	54	54	1.7
8	Time	2.6	150	968	265	377	64	618	72	71	110	3.9
	Mem	4.9	111	181	202	667	28	231	84	54	54	1.6
16	Time	1.6	79	493	147	266	38	337	40	39	60	2.2
	Mem	4.9	111	181	202	667	30	231	84	54	54	1.6
32	Time	1.2	47	276	84	133	22	187	22	24	35	1.5
	Mem	4.9	111	181	202	667	33	231	84	54	54	1.6

the 32 processors. The speedups for Algorithm I rang from a low of 13.7 for Problem A to around 24 for Problems B, E and J for 32 processors. The small size of Problems A and K limit their speedups.

Algorithm II is considerably more frugal in terms of memory than Algorithm I. Moreover, the memory required scaled down almost linearly when the number of processors increased, e.g., for Problem E it went down from about 1.1 GB for one processor to about 36.6 MB for 32 processors. The largest execution time was about 6600 seconds for Problem C with one processor but it decreased to only 3400 for 32 processors. The speedups of Algorithm II were abysmal, in the range

Table 3: Algorithm III. Execution time (seconds) and memory requirements (MB) for P=1, 2, 4, 8, 16, and 32 processors.

P	Time/ Memory	A	B	C	D	E	F	G	H	I	J	K
1	Time	11.4	1018	4803	1115	2195	367	3380	345	366	558	26.4
	Mem	8.0	180	292	325	1073	23	373	70	87	87	5.2
2	Time	6.7	516	2468	597	1125	212	1753	184	206	315	15.2
	Mem	4.6	89	147	163	538	11	187	35	44	44	5.1
4	Time	3.7	262	1290	319	582	116	914	95	114	175	8.3
	Mem	4.3	45	73	82	270	8	94	17	22	22	5
8	Time	2.5	135	719	194	313	73	520	55	68	99	5.3
	Mem	4.4	23	37	41	137	6	50	9	18	18	5.5
16	Time	1.7	70	413	116	163	49	321	34	41	63	3.7
	Mem	4.9	11	19	23	70	6	26	6	17	17	6
32	Time	0.9	39	257	72	87	28	166	19	25	36	2.5
	Mem	4.9	11	16	23	53	7	28	4	19	19	8.6

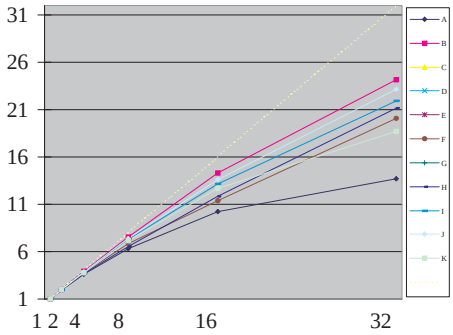

Figure 6: Algorithm I speedups.

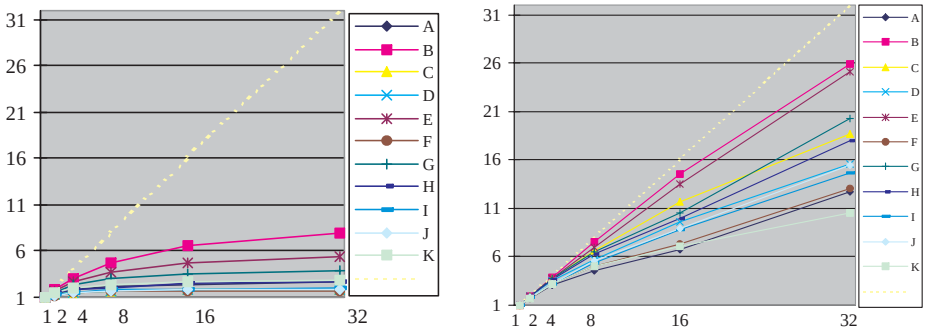

Figure 7: The speedups of Algorithms II(left) and III (right).

of 2–8 for 32 processors. For eight out of the eleven structures the speedup was less than 3 on 32

Table 4: The ratio of the execution time and the memory requirements of Algorithms II and III compared with Algorithm I

Problem	P	II/I		III/I	
		Time	Mem.	Time	Mem.
A	1	0.83	0.69	0.70	0.69
	2	1.19	0.69	0.80	0.92
	4	1.71	0.41	0.82	0.88
	8	2.46	0.22	0.96	0.90
	16	3.56	0.12	1.06	1
	32	4.25	0.06	0.75	1
B	1	0.97	0.67	0.90	0.67
	2	1.07	0.67	0.91	0.67
	4	1.24	0.41	0.91	0.41
	8	1.54	0.21	0.91	0.21
	16	2.11	0.10	0.89	0.10
	32	2.94	0.05	0.84	0.10
C	1	–	–	–	–
	2	1.41	0.67	0.69	0.67
	4	2.32	0.40	0.72	0.40
	8	3.87	0.20	0.74	0.20
	16	7.14	0.10	0.84	0.10
	32	12.38	0.06	0.93	0.09
F	1	0.98	0.83	0.85	0.83
	2	1.56	0.42	0.95	0.42
	4	2.58	0.20	1	0.28
	8	4.17	0.10	1.13	0.22
	16	6.64	0.05	1.29	0.20
	32	11.33	0.03	1.28	0.20
I	1	0.87	0.67	0.70	0.67
	2	1.35	0.67	0.79	0.67
	4	2.16	0.40	0.84	0.40
	8	3.58	0.20	0.96	0.33
	16	5.91	0.10	1.05	0.32
	32	9.48	0.05	1.05	0.35
J	1	0.86	0.67	0.69	0.67
	2	1.36	0.67	0.76	0.67
	4	2.25	0.40	0.83	0.40
	8	3.78	0.20	0.90	0.33
	16	6.42	0.10	1.05	0.32
	32	10.50	0.05	1.03	0.35

nodes.

Algorithm III seems the most balanced. Indeed, its memory requirements scaled down almost linearly with the number of processors and at the same time its speedups reached a maximum value of 25 for 32 processors, for Problem E.

Table 4 shows that the algorithms behave differently relative to one another, for different problems. Algorithm III is constantly better than the others in terms of execution time and speedup with the exception of Problem F where Algorithm I seems to scale slightly better. Algorithm II does not scale well in terms of execution time, it can be much as 12.4 times slower than Algorithm I for execution on 32 nodes.

Two sets of experimental data, Problems I and J, are related to the same virus, Sindbis; for the first we have a set of 389 views and for the second we have 643, an increase of about 65%. The corresponding increase in the execution time using 32 processors is about 48% for Algorithm I and 44% for Algorithm III. Thus both algorithms scale well in terms of the number of views. This is very important because the number of views necessary for the reconstruction of an asymmetric object is almost two orders of magnitude larger than the one for an object with icosahedral symmetry, as discussed in Section 3.

6 Orientation Refinement

We now discuss step B of the structure determination, i.e., the orientation refinement. The method used to determine the orientation of a view of a virus particle extracted from a micrograph is to project the current electron density at different orientations and then compare the experimental image with the calculated views. The unknown orientation θ, ϕ, ω, see Figure 8, is then determined to be the orientation of the calculated view of the best fit.

When the symmetry is known, the search process for orientation determination is restricted to a relatively small angular domain, see Figure 8. We consider the more challenging case when the information regarding the symmetry of the virus particle is not available. The advantage of the method described in this paper is that one could use it not only to study the structure of the protein shell, but also the structure of the genome. Moreover, if the virus exhibits any symmetry this method allows us to determine the symmetry group.

Previous parallel orientation refinement programs such as the one reported in [Joh94] took advantage of the embarrassingly parallel nature of the traditional algorithm. Our algorithm is radically different, it does not make any assumption about the symmetry of the object, it constructs cuts in the electron density map, it is based upon a multi-resolution search, and a sliding window mechanism discussed for the first time in [Bak97]. The orientation refinement algorithm is used in conjecture with our `3D` reconstruction algorithm in Cartesian coordinates for objects without symmetry [Lyn99, MaY01, MaY02].

6.1 Problem Formulation

We use the following terms and notations:

• D is the current version of the electron density map of size l^3.

• $\mathcal{D} = DFT(D)$ is the `3D` Discrete Fourier Transform (DFT) of the electron density map.

• $\mathcal{C}$ is a set of `2D` cuts of $\mathcal{D}$ by interpolation in the `3D` Fourier domain .

• $\mathcal{E} = \{\mathcal{E}^1, \mathcal{E}^2, \ldots \mathcal{E}^q, \ldots \mathcal{E}^m\}$ with $1 \leq q \leq m$ is the set of m experimental views; each view is of size $l \times l$ pixels.

• $\mathcal{F} = \{\mathcal{F}^1, \mathcal{F}^2, \ldots \mathcal{F}^q, \ldots \mathcal{F}^m\}$ is the set of `2D` DFTs of experimental views. Here $\mathcal{F}^q = DFT(\mathcal{E}^q)$ for $1 \leq q \leq m$.

• $O_{init} = \{O^1_{init}, O^2_{init}, \ldots, O^q_{init} \ldots, O^m_{init}\}$ is the set of initial orientations, one for each view. $O^q_{init} = \{\theta^q_{init}, \phi^q_{init}, \omega^q_{init}\}$

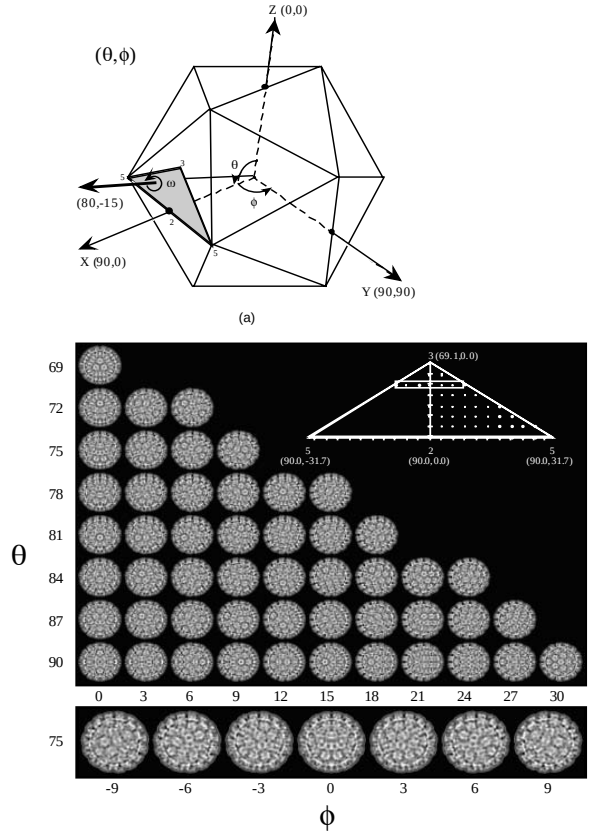

Figure 8: (a) The three angles used to characterize the orientation of a view. (b) The set of calculated views for an icosahedral virus at 3 degree resolution. Courtesy of T. Baker.

• $O_{refined} = \{O^1_{refined}, O^2_{refined}, \ldots, O^q_{refined} \ldots, O^m_{refined}\}$ is the set of refined orientations, one for each view. $O^q_{refined} = \{\theta^q_\mu, \phi^q_\mu, \omega^q_\mu\}$

• P is the number of nodes available for program execution. Note that parallel I/O could reduce the I/O time but in our algorithm we do not assume the existence of a parallel file system. To avoid contention a coordinator node typically reads an entire data file and distributes data segments to the nodes as needed.

• Given a `3D` lattice D of size l^3 we define a z-slab of size $zslab_{size}$ to be a set of consecutive $zslab_{size}$ xy-planes. One can similarly define x- and y-slabs.

• The resolution of the electron density map is denoted by r_{map}. In the reconstruction process we use only the Fourier coefficients up to $\frac{1}{r_{map}}$.

• The angular resolution is denoted by $r_{angular}$.

Given: (1) a set of m views and (2) the electron density map, the goal is to find out the orientation of each view.

Several methods including the method of "common lines", [Bak99] can be used to this end. Here we describe a procedure for the refinement of orientations that is less sensitive to the noise caused by experimental errors. The basic idea is to project the electron density at known angles and then to compare each experimental view with the calculated view. Once we define the distance between

an experimental view and a calculated view, the goal of the search is to identify the calculated view at the minimum distance from the experimental view. The procedure does not make any assumptions about the symmetry, but can detect symmetry if one exists.

We are only concerned here with the orientation refinement; in other words we are given a rough estimation of the orientation, say at 3 degree angular resolution and our goal is to reach a resolution of say 0.01 degree or better.

This paper describes an algorithm where the search is conducted in the Fourier domain. We first perform a 2D Discrete Fourier Transform (DFT) of the experimental view, $\mathcal{F}$, apply a CTF correction to it, and then compare it with a cut at a precise orientation through the 3D DFT of the electron density map, $\mathcal{C}$.

In this algorithm the distance between two $l \times l$ arrays of complex numbers:
$\mathcal{F} = [a_{j,k} + ib_{j,k}]_{1 \leq j,k \leq l}$, and $\mathcal{C} = [c_{j,k} + id_{j,k}]_{1 \leq j,k \leq l}$, with $i = \sqrt{-1}$.
is defined as:

$$d(\mathcal{F}, \mathcal{C}) = \frac{1}{l^2} \sum_{j=1}^{l} \sum_{k=1}^{l} \sqrt{([a_{j,k} - c_{j,k}]^2 + [b_{j,k} - d_{j,k}]^2)}.$$

To determine the distance, $d(\mathcal{F}, \mathcal{C})$ at a given resolution we use only the Fourier coefficients up to $\frac{1}{r_{map}}$; thus, the number of operations is reduced accordingly.

The algorithm is embarrassingly parallel, each experimental view can be processed independently by a different processor. But the 3D electron density map and its DFT can be very large; the database of calculated views could require several TBytes of data and the electron density may need several Gbytes of storage.

The size of the search space $\mathcal{S}$ is very large; if the initial orientation of an experimental view, $\mathcal{E}^q$ is given by $O^q = \{\theta^q, \phi^q, \omega^q\}$ then the cardinality of the set $\mathcal{S}$ is:

$$|\mathcal{S}| = \frac{\theta^q_{max} - \theta^q_{min}}{r_{angular}} \times \frac{\phi^q_{max} - \phi^q_{min}}{r_{angular}} \times \frac{\omega^q_{max} - \omega^q_{min}}{r_{angular}}.$$

This step requires $\mathcal{O}(l^2 \times |\mathcal{S}|)$ arithmetic operations. For example, if $r_{angular} = 0.1$ degree and the search range is from 0 to π for all three angles, then the size of the search space is very large: $|\mathcal{S}| = (1800)^3 = 5.832 \times 10^9$.

Figure 8 (b) shows that the corresponding size for an icosahedral particle at 3 degree angular resolution consists of only 51 calculated views; at 0.1 degree the size of the search space is less than $4,000$ calculated views, [Bak99]. Thus, for an asymmetric particle the size of the search space increases by *six (6) orders of magnitude* compared with an icosahedral particle !! Moreover, when comparing two views of an icosahedral virus particle, a calculated and an experimental one, we could use only a shell of thickness corresponding to the capsid, rather than the entire 2D image.

6.2 A Sliding Window Multi-Resolution Parallel Orientation Refinement Algorithm

The orientation refinement is a *multi-resolution* process. Typically, we carry out several refinement steps at different angular resolutions, e.g., one at $r_{angular} = 1$ degreefollowed by one at $r_{angular} = 0.1$ degree, one at $r_{angular} = 0.01$ deg, and finally one at $r_{angular} = 0.002$ degree The advantage of this approach is clearly illustrated by an example: assume that the initial value is say $\theta = 65$ deg, the search domain is 60 to 70 degreeand we require an angular resolution of 0.002 degree A one step search would require 5000 matching operations versus 35 for a multi-resolution matching. The

orientation of an experimental view is given by three angles (θ, ϕ, ω), therefore the multi-resolution approach reduces the number of matching operations for a single experimental view by almost four orders of magnitude. As pointed out earlier, several thousand experimental views are needed for the reconstruction of the `3D` electron density map at a high resolution. A matching operation consists of two steps (1) construct a cut into $\mathcal{D}$ with a given orientation and (2) compute the distance between the `2D` DFT of the experimental view, $\mathcal{F}^q$, and the cut.

The search domain at a given angular resolution is extended whenever the best fit occurs near the edge of the search domain. This *sliding-window* approach increases the number of matching operations, but, at the same time improves the quality of the solution; it guarantees that the true optimal match is found. A similar strategy is used for refining the centers of the experimental images.

The orientation refinement algorithm consists of the following steps:

Step a. Construct $\mathcal{D}$, the `3D` Discrete Fourier Transform, DFT, of the electron density map.

(a.1) The coordinator reads all **z**-slabs of the entire electron density map D.

(a.2) The coordinator sends to each other node a **z**-slab of the electron density map D of size $t_{slab} = \lceil \frac{l}{P} \rceil$.

(a.3) Each node carries out a `2D` DFT calculation along the **x**- and **y**-directions on its **z**-slab.

(a.4) A global exchange takes place after the `2D` DFT calculation and each node ends up with a **y**-slab of size t_{slab}.

(a.5) Each node carries out a `1D` DFT along the **z**-direction in its **y**-slab.

(a.6) Each node broadcasts its **y**-slab.

After the `all-gather` operation each node has a copy of the entire $\mathcal{D}$. To perform calculations at resolution r_{map} we only keep a subset of the $\mathcal{D}$, within a sphere of radius $\frac{1}{r_{map}}$.

This step requires a total of $\mathcal{O}(l^3 \times log_2(l^3))$ arithmetic operations and $\mathcal{O}(l^3)$ words of memory in each node.

Step b. Read in groups of $m' = \lceil \frac{m}{P} \rceil$ views, $\mathcal{E}$, from the file containing the `2D` views of the virus, and distribute them to all the processors. The amount of space required to store the experimental views on each processor is: $m' \times (b \times l^2)$ with b the number of bytes per pixel. In our experiments $b = 2$.

Step c. Read the orientation data file containing the initial orientations of each view, $O_{init} = \{O^1_{init}, O^2_{init}, \ldots, O^q_{init} \ldots, O^m_{init}\}$. Distribute the orientations to processors such that a view $\mathcal{E}^q$ and its orientation O^q_{init}, $1 \leq q \leq m$, are together.

At a given angular resolution we perform the following operations for each experimental view $\mathcal{E}^q$, $1 \leq q \leq m$:

Step d. Compute $\mathcal{F}^q$, the `2D` DFT of experimental views. Each processor carries out the transformation of the m' views assigned to it.

This step requires $\mathcal{O}(l^2 \times log_2(l^2))$ arithmetic operations for each experimental view and $\mathcal{O}(l^2)$ words of memory for the data.

Step e. Perform the CTF-correction of the DFT of each view $\mathcal{F}^q$, $1 \leq q \leq m$. Use the CTF-factor, $\tau^q = \tau^q_1 + i\tau^q_2$ to correct every Fourier coefficient $(a^q_{j,k}, b^q_{j,k})$ of $\mathcal{F}^q$ and obtain a new value $(\alpha^q_{j,k}, \beta^q_{j,k})$:

$$\alpha^q_{j,k} + i\beta^q_{j,k} = (a^q_{j,k} + ib^q_{j,k}) \times (\tau^q_{1,j,k} + i\tau^q_{2,j,k}), \quad 1 \leq (j,k) \leq l, \quad 1 \leq q \leq m.$$

Note that the views originated from the same micrograph have the same CTF-factor, τ^q. This step requires $\mathcal{O}(l^2)$ operations for each experimental view.

Step f. Given:
(i) the experimental view $\mathcal{E}^q$ with the initial orientation $O^q_{init} = \{\theta^q_{init}, \phi^q_{init}, \omega^q_{init}\}$,
(ii) the search domain,

$$\begin{aligned} 0 \leq \theta^q_{min} \leq \theta^q_s \leq \theta^q_{max} \leq \pi, \\ 0 \leq \phi^q_{min} \leq \phi^q_s \leq \phi^q_{max} \leq \pi, \\ 0 \leq \omega^q_{min} \leq \omega^q_s \leq \omega^q_{max} \leq \pi \end{aligned}$$

construct a set of $2D$-cuts of $\mathcal{D}$, the 3D-DFT of electron density map by interpolation in the 3D Fourier domain. Call

$$\mathcal{C}^q = \{\mathcal{C}^q_1, \mathcal{C}^q_2, \ldots \mathcal{C}^q_s, \ldots \mathcal{C}^q_w, \}$$

the set of planes spanning the search domain for $\mathcal{E}^q$. Call $O^q_s = \{\theta^q_s, \phi^q_s, \omega^q_s\}$ the orientation of the cut $\mathcal{C}^q_s$.

Call w the number of calculated cuts in $\mathcal{D}$ for a given angular search range and angular resolution. $w = w_\theta \times w_\phi \times w_\omega$. Typical values are $w_\theta = w_\phi = w_\omega = 10$, thus $w = 1000$.

This step requires $\mathcal{O}(w \times l^2)$ arithmetic operations for each experimental view.

Step g. Determine the distance of $\mathcal{F}^q$ to every $\mathcal{C}^q_s$, $d^q_s = d(\mathcal{F}^q, \mathcal{C}^q_s)$ for $1 \leq s \leq w$.

This step requires $\mathcal{O}(w \times l^2)$ arithmetic operations for each experimental view.

Step h. Compute the minimum distance:

$$d^q_\mu = min\{d^q_1, d^q_2, \ldots, d^q_w\}.$$

Call O^q_μ the orientation of the cut $\mathcal{C}^q_\mu$,

$$O^q_\mu = \{\theta^q_\mu, \phi^q_\mu, \omega^q_\mu\}.$$

This step requires $\mathcal{O}(w)$ arithmetic operations for each experimental view.

Step i. If any of the three angles corresponding to the minimum distance cut, $O^q_\mu = (\theta^q_\mu, \phi^q_\mu, \omega^q_\mu)$ is at the edge of the original search domain defined in step (f), redefine the search domain. Make O^q_μ the center of the new search domain, and repeat steps (f), (g), and (h).

Call n_{window} the number that we slide the window. Then the total number of operations required for each view in steps (f), (g), and (h) is: $\mathcal{O}(n_{window} \times w \times l^2)$.

Step j. Assign to experimental view $\mathcal{E}^q$ the orientation of the minimum distance cut, O^q_μ.

Step k. Refine the position of the center of the 2D DFT. Move the center of $\mathcal{E}^q$, $(x^q_{center}, y^q_{center})$ within a box of size $2\delta_{center}$ using the current orientation and determine the best fit with the minimum distance cut $\mathcal{C}^q_\mu$.

For each new value of the center $(x^q_{center,i}, y^q_{center,i})$, determine the distance to $\mathcal{C}^q_\mu$:

$$d^q_{\mu,i} = d(\mathcal{E}^q_i, \mathcal{C}^q_\mu)$$

Find the minimum distance:

$$d^q_{\mu,opt} = min\{d^q_{\mu,1}, d^q_{\mu,2}, \ldots d^q_{\mu,n_{center}}\}$$

where n_{center} is the number of center locations considered. For example, if we use a 3×3 box $n_{center} = 9$.

If the $(x^q_{center,opt}, y^q_{center,opt})$ is at the edge of the search box redefine the search box. Make $(x^q_{center,opt}, y^q_{center,opt})$ the center of a new search box, and repeat step k. Call n_{window} the number that we slide the window. Then, the total number of operations required for each view in this step is: $\mathcal{O}(n_{center} \times n_{window} \times l^2)$.

Step l. Correct $\mathcal{E}^q$ to account for the new center.

Step m. Wait for all nodes to finish processing at a given angular resolution.

Step n. Repeat the computation for the next angular resolution until the final angular resolution is obtained. Then:
$O_{refined} = \{O^1_{refined}, O^2_{refined}, \ldots, O^q_{refined} \ldots, O^m_{refined}\}$ is the set of refined orientations, one for each view. Here $O^q_{refined} = \{\theta^q_\mu, \phi^q_\mu, \omega^q_\mu, x^q_{center,opt}, y^q_{center,opt}\}$
Step o. Write out the refined orientation file.

As pointed out in Section 6.1 the structure determination is an iterative process. Given a resolution r_{map} we use the refined orientations and the new centers to reconstruct the electron density map. The new map is used again for another step of orientation refinement and the process continues until we cannot further refine the structure at that particular resolution. Then we increase the resolution and repeat the entire procedure.

6.3 Experimental Results

Our objectives were threefold: (a) to verify the correctness of our results, (b) to determine our ability to increase the resolution of the structure determination using the new algorithms, and (c) to obtain some indication about the performance of our programs.

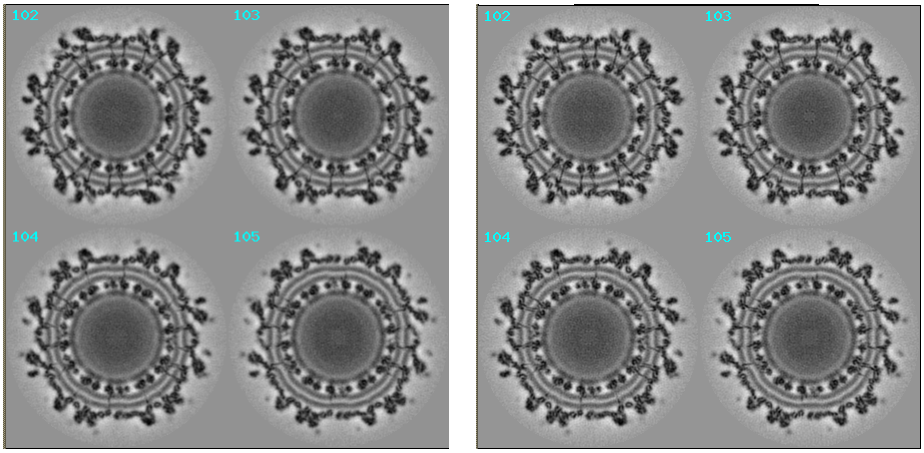

Figure 9: Cross sections 102-105 of electron density map of the Sindbis virus obtained with its old orientation (left) and the orientation produced by our new algorithm (right).

We present results regarding two virus structures, those of Sindbis and reo viruses. The experimental data for the Sindbis virus consists of 7,917 views of 221 × 221 pixels each; the highest resolution obtained using existing algorithms is 11.2 Å. The experimental data for the reo virus consists of 4,422 views of 511 × 511 pixels each; the highest resolution obtained using existing algorithms is 8.6 Å.

We use our new algorithm to refine the orientation of these data for angular resolutions $r_{angular}$ = 1 deg, 0.1 deg, 0.01 degreeand 0.002 deg, and with center resolution δ_{center} = 1 pixel, 0.1 pixel,

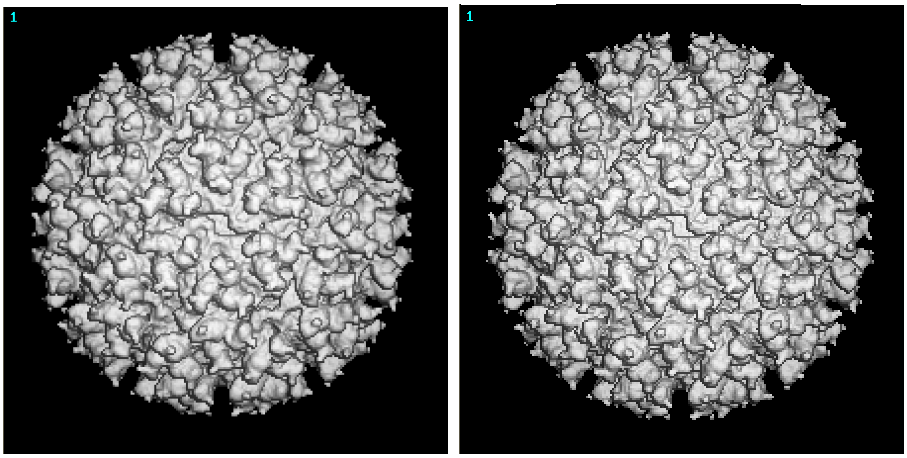

Figure 10: 3D electron density map of the Sindbis virus obtained with its old orientation (left) and our new refined orientation(right).

0.01 pixel and 0.002 pixel. In our calculations we reached a resolution of 10.0 Å for the Sindbis virus and 8.0 Å for the reo virus, see Figures 12 and 13.

The first objective was achieved by comparing our results with those obtained using existing algorithms. Figure 9 shows the cross sections 102-105 of the 3D electron density map of Sindbis virus, reconstructed respectively with its old orientation and with our new refined orientation. Figure 10 shows the 3D density map of Sindbis virus, reconstructed respectively with its old orientation and with our new refined orientation. Though differences in the two maps are difficult to visualize directly in low magnification views, high magnification views do reveal more details in the new density map.

To test the resolution achieved, we use the procedure illustrated in Figure 11. After the last step of the orientation refinement at a given resolution we compute two 3D reconstructions, one using only odd numbered experimental views and the other, even numbered views. Then we determine the correlation between the two maps. Figure 12 shows a plot of the correlation coefficients for the new reconstructed maps compared with the one based upon previously determined orientations for the Sindbis virus. Figure 13 shows a plot of the correlation coefficients for the new reconstructed maps compared with those based upon previously determined orientations for the reo virus.

Figures 12 and 13 indicate that the new orientation refinement method gives higher correlation coefficients and hence enable us to construct electron density maps at higher resolution.

To determine the highest resolution we examine the plot of the correlation coefficient, cc, versus resolution and determine the crossing point of the graph and the $cc = 0.5$ line. A correlation coefficient higher than 0.5 gives a conservative estimate of the final resolution of the entire density map. Figure 12 shows that the graph corresponding to the new method crosses the $cc = 0.5$ line at 10.0 Å versus 11.2 for the old method. Figure 13 shows that the correlation coefficient for new method is 0.5 at 8.0 Å while the one for the old method does the same at 8.7 Å .

Finally, we evaluate the performance of our new algorithms on the same IBM SP2 system. Tables 5 and 6 show the evolution of the minimum distance, the orientation, and the center for one experimental view of Sindbis and one of reo viruses. The data shows that both the orientation and the position of the center are improved when the angular resolution increases from 1, to 0.1, then to 0.01, and finally to 0.002 degree.

Tables 7 and 8 show the time for different steps of the orientation refinement process for one

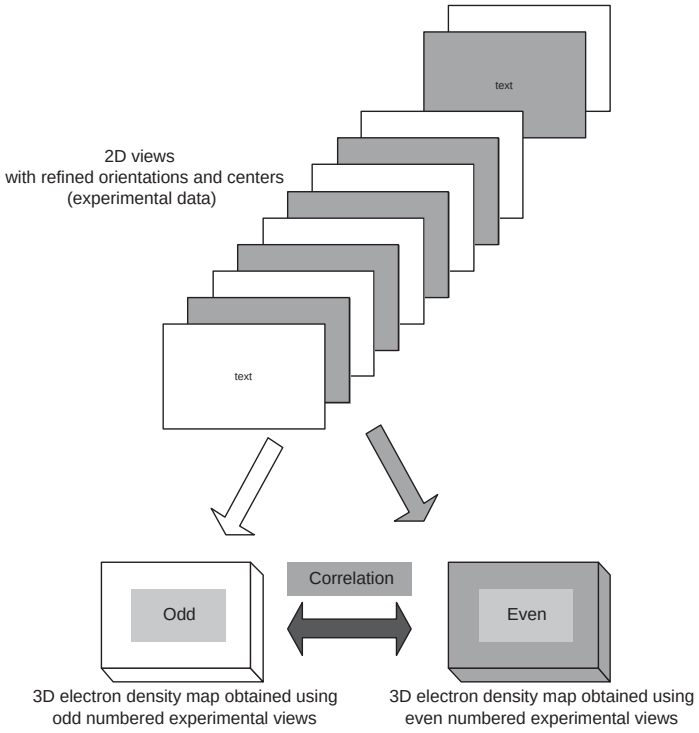

Figure 11: The procedure used to determine the resolution of the electron density map.

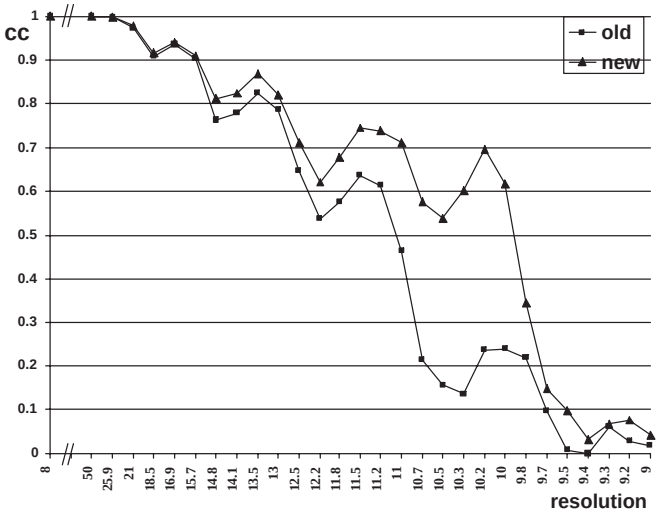

Figure 12: Correlation-coefficient plot for Sindbis virus.

iteration of structure determination for the Sindbis and the reo virus. Tables 7 and 8 show that 99% of the time for ordination refinement, is spent in matching. The `3D` reconstruction time represents less than 5% of the total time per cycle if an equal number of processors would have been used for both steps. Moreover we see in some instances the sliding window mechanism activated, for

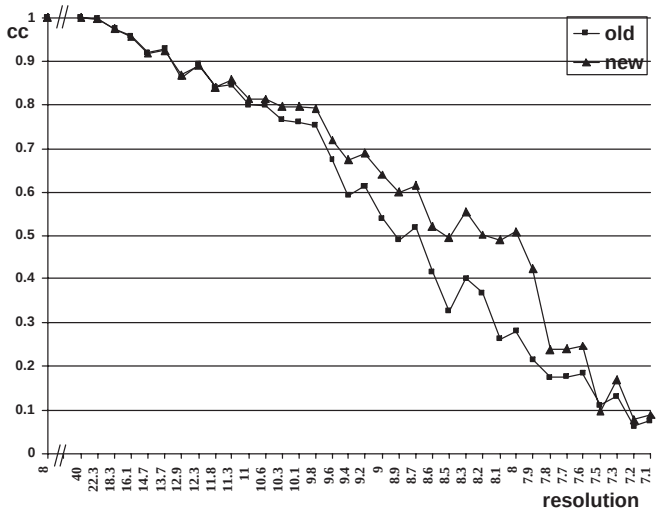

Figure 13: The correlation-coefficient plot for reo virus.

Table 5: The effect of angular resolution orientation upon the minimum distance, the three orientation angles (θ, ϕ, ω) and the center of a view (x_{center}, y_{center}) for one view of Sindbis virus

Angular resolution	Minimum distance	θ	ϕ	ω	x_{center}	y_{center}
1 degree	1166375.375	77.245	2.036	354.344	109.285	108.978
0.1 degree	1166322.000	77.345	1.936	354.144	109.385	108.878
0.01 degree	1166283.625	77.395	1.866	353.994	109.415	108.888
0.002 degree	1166282.500	77.389	1.864	353.986	109.405	108.886

Table 6: The effect of angular resolution orientation upon the minimum distance, the three orientation angles (θ, ϕ, ω) and the center of a view (x_{center}, y_{center}) for one view of the reo virus

Angular resolution	Minimum distance	θ	ϕ	ω	x_{center}	y_{center}
1 degree	7363170.5	83.144	1.425	101.586	256.821	260.672
0.1 degree	7363138.0	83.044	1.425	101.586	256.721	260.672
0.01 degree	7363127.5	83.094	1.435	101.616	256.741	260.652
0.002 deggree	7363124.0	83.092	1.431	101.614	256.741	260.652

example, at 0.01 degree instead of 9 matchings we needed 15 for the Sindbis virus and 11 for the reo virus. The time and memory requirements for the orientation refinement and for the `3D` reconstruction are shown in Figure 14. Our orientation refinement program runs on 16 processors and our parallel 3D reconstruction 8 processors for these two data sets. The execution time for `3D` reconstruction for the Sindbis virus on 8 processors is 9,150 seconds and for the reo virus is 15,866 seconds.

Table 7: The time for different steps of the orientation refinement process for the structure determination of Sindbis virus.

Angular resolution (degree)	Search range	3D DFT (sec)	Read image (sec)	FFT analysis (sec)	Orientation refinement (sec)	Total time (sec)
1	9	19	246	46	14,053	14,364
0.1	9	15	152	46	14,109	14,308
0.01	15	13	158	46	71,065	71,282
0.002	9	13	155	46	26,901	27,116

Table 8: The time for different steps of the orientation refinement process for the structure determination of reo virus.

Angular resolution (degree)	Search range	3D DFT (sec)	Read image (sec)	FFT analysis (sec)	Orientation refinement (sec)	Total time (sec)
1	9	175	550	138	19,942	20,805
0.1	9	206	533	142	21,957	22,839
0.01	11	178	573	137	69,672	70,561
0.002	9	155	529	138	43,786	44,608

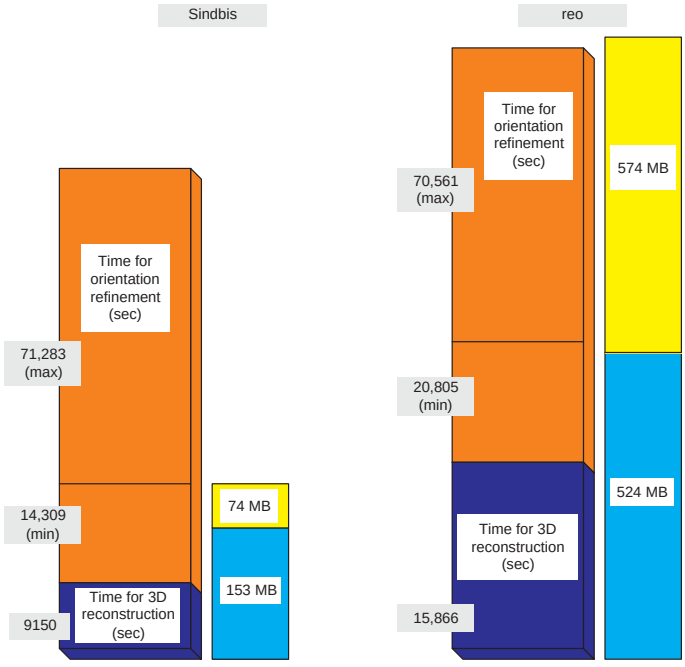

Figure 14: The time per iteration and the memory requirements for the orientation refinement and for the 3D reconstruction for the Sindbis virus (left) and for the reo virus (right)..

7 An Environment for Computational Structural Biology

In addition to efficient algorithms for structure refinement we need an intelligent computing environment to support structural biology studies. Even from the brief description presented in Section 4 it is abundantly clear that computing is only one facet of the complex task to determine the structure of a virus; preparing the sample to be irradiated in the microscope, understanding the physical limitations of a complex measuring instrument (the electron microscope), gathering the data, evaluating the errors, and above all evaluating the results of each step, are very challenging tasks. It is unreasonable to expect a structural biologist to fully understand the intricacies of the algorithms and the subtleties of parallel computing and parallel systems. Therefore, we need an intelligent environment capable to make decisions based upon a set of rules and a knowledge base summarizing important performance data from past experiments.

Consider for example the problem of algorithm selection. We have already seen that data intensive applications such as the one presented in Section 5 require algorithms capable of adapting to the problem size, to the architecture, the configuration, and the amount of resources of a target parallel system. If a system has limited memory, a high latency and low bandwidth interconnection network, and very fast processors we might chose to replicate some computations in each node. Such space-time tradeoffs are common in the design of efficient parallel algorithms.

Another function expected from the environment is to select parameters of the algorithm a layperson may have a hard time relating to. For example, consider the zero-fill aspect ratio discussed in Section 5. As discussed in our earlier work, the larger is the value of k, the lower are the interpolation errors. The intuitive justification for this effect comes from the sampling theorem [Gon93]; increasing k decreases the size of the sampling interval in the reciprocal space (Fourier transform domain). Yet, increasing k increase the amount of memory used by the `3D` reconstruction algorithm. Thus another rule is necessary: select the largest value of k that can be tolerated by the target system configuration.

Last but not least, decisions regarding the stopping condition at a given resolution and the need to increase the resolution based upon the study of the correlation coefficients can be automated. Section 6.3 provides the insights into this process.

7.1 Coordination

To design an intelligent environment we have to decompose the complex task of structure refinement into a number of simpler activities and then to coordinate the execution of these activities. Figure 15 presents a high level view of the environment we are currently constructing.

In Figure 15(a) we see the major steps required by structure refinement; (1) problem analysis, (2) resource discovery and reservation, (3) initial setup of the execution environment, and (4) coordination of various execution tasks.

A control agent interacts with the experimentalist on one hand and with a group of agents managing each of the steps described earlier. The problem analysis is the responsibility of an agent capable to parse the problem description file. The agent extracts information such as: the number of views, the size of each view, the symmetry if known, the initial resolution, the final resolution desired, the physical characteristics of the images collected. Then it creates a knowledge base record of the problem description.

The functions of the resource discovery and reservation agent cover actions ranging from interacting with a batch queuing system to a full blown resource discovery in a computational grid. Broker agents, mediator agents and other societal agents provided by the grid infrastructure may

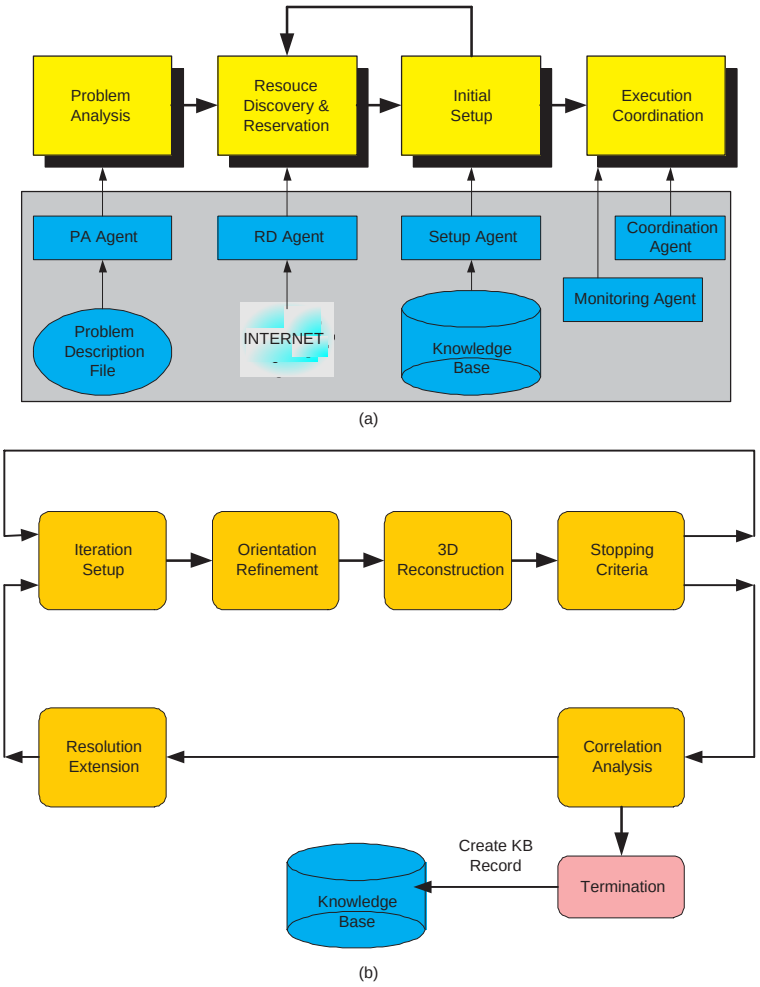

Figure 15: (a) The sequence of steps for structure refinement. (b) The actions taken during the execution coordination phase.

be involved in this process [Mar02].

The initial setup requires an analysis of the current problem description record, of the description and performance records of past problems to determine the minimum as well as the optimum resource requirements. There is a constant interaction of this agent with the resource reservation agent. At this stage "hidden" parameters such as the zero-fill aspect ratio are determined.

Finally, the execution coordination agent carries out the tasks illustrated in Figure 15(b). It controls individual iterations consisting of orientation refinement and `3D` reconstruction. At the end of each iteration it checks the stopping criteria and decides to carry out an additional iteration if an improvement of the electron density map is noticeable or it requires a correlation analysis. As a result of the correlation analysis we may continue at the same resolution, the resolution may be extended, or we may decide to backtrack to a lower resolution.

7.2 The Agents

The agents are written using the Blueprint agent description language and run in the Bond environment [BoJ00]. Bond is a Java based agent framework available as a open source under an LGPL license from our web site, **http://bond.cs.ucf.edu**. The system has been used in the past to build a network of PDE solvers, [BoM00], and for several other scientific applications.

The control agent runs on a workstation, PC, a laptop, or possibly a wireless mobile device connected to the Internet. Once started, each parallel program runs under the control of a monitoring agent. Upon completion, the monitoring agent scans the output files of the parallel program and produces a performance execution record and stores it in the knowledge base. The knowledge base is implemented as a shared data space using an IBM T-Spaces server integrated with the Bond environment. This implementation allows our problem solving environment to work well with any distributed system accessible via the Internet. The monitoring agent may also send the output data to a graphics device for visualization or may simply send it to the persistent storage server.

If any of the nodes involved in the computation fails the resident monitoring agent is no longer capable to send periodic updates to the coordinator. The coordinator is also responsible to keep partial results so that the computation could be restarted or we may backtrack.

Agents are mobile and run in the proximity of entities they interact with. This strategy allows us to minimize communication costs; for example, a monitoring agent runs at the execution site, the coordinator agent runs at a site close to all execution sites and the control agent runs on the network access device.

8 Summary, Recent Results, and Future Work

We discuss the application of parallel and distributed computing to structural biology, in particular to 3D reconstruction of viruses from electron micrographs. The number of views/projections required for the reconstruction of an asymmetric object is almost two orders of magnitude larger than the one for an object with icosahedral symmetry. Moreover, the structural biologists wish to study larger structures including complexes of viruses and antiviral agent, and to increase the resolution of the reconstruction to about 5Å. These factors make a data intensive problem even more demanding in terms of CPU cycles, memory requirements, communication, and I/O bandwidth.

First, we present three parallel algorithms for 3D reconstruction of an asymmetric object from its 2D views, specify their resource requirements, and present results of performance measurements. These algorithms have several desirable features. They allow the reconstruction of asymmetric objects, thus they can be used to study the genetic material of the virus core. They scale well in terms of the number of views used for 3D reconstruction. For more than thirty years the structural biology community has used variations of one of the five algorithms proposed in [Cro70], with the 3D reconstruction being done with Fourier-Bessel transforms. Our algorithms use Cartesian coordinates, are easily parallelizable, and use interpolation instead of least squares as is suggested in [Cro70]. At the time of this writing, we are not aware of any program for the reconstruction of a 3D asymmetric object, of the size and at the resolution we deal with, when the 3D electron density map is obtained from an arbitrary, sufficiently large, set of 2D views of the object.

We also present a novel algorithm for the refinement of orientations of individual virus particle views for asymmetric particles and analyze its computational complexity. The algorithm can be used to determine the symmetry group of a symmetric particle and for the 3D reconstruction of particles exhibiting no symmetry or any symmetry. Note that even when the symmetry group of the protein shell is known, the genetic material at the core of the cell does not exhibit any symmetry

and our algorithm can be used to study such systems.

The orientation algorithm presented in this paper was designed for a distributed memory parallel architecture. On a shared memory system we would need one copy of the electron density map and of its 3D DFT. On a distributed memory system we choose to replicate the electron density map and its 3D DFT on every node or groups of nodes shared memory because we wanted to reduce the communication costs. The alternative is to implement a shared virtual memory where 3D "bricks" of the electron density or its DFT are brought on demand in each node when they are needed, a strategy presented in [Cor93].

We tested our programs using experimental data gathered for symmetric virus particles because we wanted to compare the quality of our solution with the one produced by programs that exploit the known symmetry of the protein shell of a virus particle. Such programs have been used for many years by the structural biology community. Our results indicate that our algorithm, in addition to enabling structural biologists to study asymmetric systems, provides better quality solutions than the algorithms and programs they currently use.

Finally, we discuss an intelligent problem solving environment for structural biology. The environment is based upon the agent framework we have developed for the past four years. A new version of the Bond system is planned and this will probably allow us to increase the sophistication of our environment.

In the last few months, we increased the resolution of the Sindbis virus reconstruction from 10.0 Å to 8.5 Å (Figure 16) and of the reo virus reconstruction from 8.0 Å to 7.0 Å (Figure 17). For the Sindbis virus, we still use 7,917 views of 221×221 pixels each. The new reconstruction shows more details of the structure (Figure 18). For the reo virus, we added more images, now we use 7,937 views of 511×511 pixels each.

We continue our work on several directions: (i) test the limits of our algorithms by refining several structures at increasingly high resolution; (ii) improve the quality of the solution (e.g., use only the experimental data in a shell adapted for each resolution to reduce the level of noise); (iii) optimize the performance of our algorithms for different system configurations (we are in the process of acquiring a cluster with multiple processors per node and multiple network connections among notes, and we plan to create a version of the algorithms exploiting the additional communication bandwidth and the shared memory); (iv) create the ontologists necessary for process coordination and continue the development of our agent-based coordination architecture. The source code of our programs and some test cases are available from **http://bond.cs.ucf.edu**.

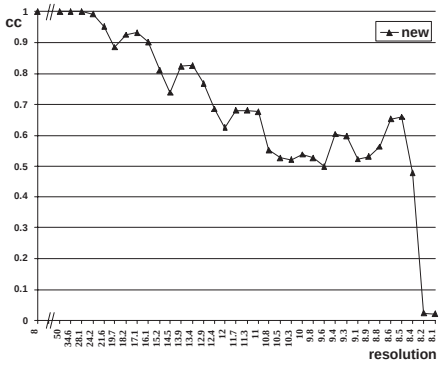

Figure 16: Correlation coefficient plot for the Sindbis virus at 8.5 Å resolution.

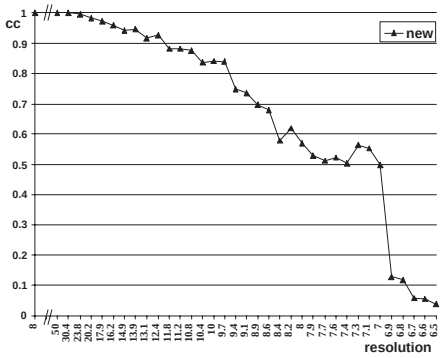

Figure 17: Correlation coefficient plot for the reo virus at 7.0 Å resolution.

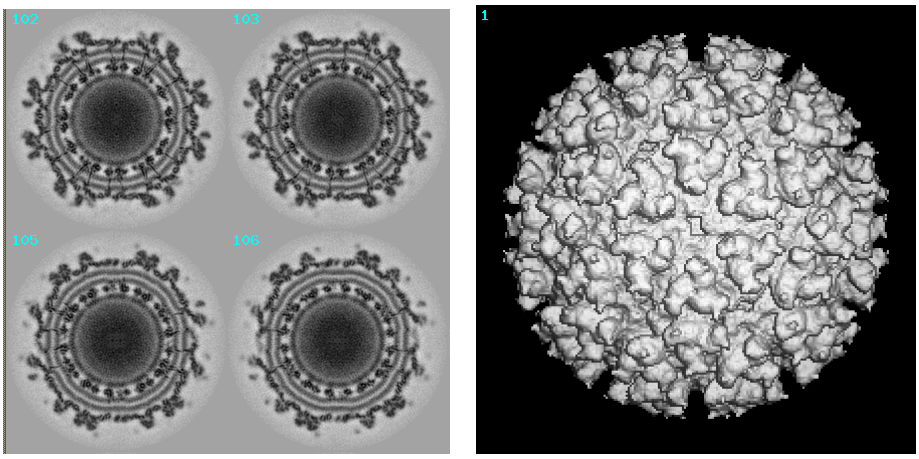

Figure 18: Cross sections 102-105 (left) and 3D electron density map (right) of the electron density map of the Sindbis virus at 8.5 Å resolution.

9 Acknowledgments

We are much indebted to our colleagues from the Structural Biology Group at Purdue University. Timothy S. Baker, a leading authority in cryo-TEM, motivated our work, contributed many new ideas, and worked very closely with us for the past decade. His group including Wei Zhang and Xiaodong Yan, shared their data with us. Insights to structural biology were provided by the distinguished crystallographer and structural biologist, Michael G. Rossmann.

Many colleagues who joined our group along the years made significant contributions. Robert E. Lynch contributed many important ideas to the `3D` reconstruction algorithm, Hong Lin optimized several functions of the program. Ioana Martin contributed the automatic particle selection method. Lotzi Boloni is a major contributor to the Bond system. Magdalena Marinescu's suggestions helped improve the quality of this paper.

The research reported in this paper was partially supported by National Science Foundation grants MCB9527131, DBI0296107, ACI0296035, and EIA0296179.

References

[Bak88] Baker, T. S., J. Drak, and M. Bina, "Reconstruction of the three-dimensional structure of simian virus 40 and visualization of chromatin core", Proc. Natl. Acad. Sci. USA, 85:422-426, 1988.

[Bak96] Baker, T.S. and R.H. Cheng. " A model-based approach for determining orientations of biological macromolecular electron microscopy". Journal of Structural Biology, 116, pp. 120-130, 1996.

[Bak97] Baker, T. S., I. M. B. Martin, and D. C. Marinescu. "A parallel algorithm for determining orientations of biological macromolecules imaged by electron microscopy". CSD-TR #97-055, Department of Computer Sciences, Purdue University, 1997.

[Bak99] Baker, T. S., N. H. Olson, and S. D. Fuller. "Adding the third dimension to virus life cycles: Three-dimensional reconstruction of icosahedral viruses from cryo-electron microscopy", Microbiology and Molecular Biology Reviews, Vol. 63, No. 4, pp. 862-922, 1999.

[BoJ00] Bölöni L., K. K. Jun, K. Palacz, R. Sion, and D. C. Marinescu, "The Bond agent system and applications", in *Agent Systems, Mobile Agents, and Applications,* (D. Kotz and F. Mattern, eds.), Lecture Notes on Computer Science, Vol. 1882, Springer Verlag, pp. 99–112, 2000.

[BoM00] Bölöni L., D. C. Marinescu, J. R. Rice, P. Tsompanopoulu, and E. A. Vavalis, "Agent-based scientific simulation and modeling", Concurrency Practice and Experience, Vol. 12, pp. 845-861, 2000.

[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), Vol. 386, pp. 88–91, 1997.

[BrR67] Bracewell, R. H., and A. C. Riddle, "Inversion of fan beam scans in radio astronomy", Astrophysics Journal, Vol. 150, pp. 427 - 434, 1967.

[Bra91] C. Branden and J. Tooze, "Introduction to protein structure", Garland Publishing, New York and London, 1991.

[Bri96] Bricogne, G., and Irwin, J., "Maximum-likelihood structure refinement: theory and implementation within BUSTER+TNT", in Molecular Refinement: Proc CCP4 Study Weekend, January 1996, Dodson E., Moore M., Ralph, A., and Bailey S., (Eds). CCLRC Darebury Laboratory, pp. 85-92, 1996.

[Che95] Cheng, R.H., R. J. Kuhn, N. H. Olson, M. G. Rossmann, H.-K. Choi, T. J. Smith, and T. S. Baker. "Three-dimensional structure of an enveloped alphavirus with T = 4 icosahedral symmetry", Cell, 80, 621-630, 1995.

[Cor93] Cornea-Hasegan, M. A. D. C. Marinescu, Z. Zhang, J. R. Rice, R. E. Lynch, and M. G. Rossmann, "Macromolecular electron density averaging on distributed memory MIMD systems", Concurrency: Practice and Experience, Vol. 5, No. 8, pp. 635-657, 1993.

[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), Vol. 386, pp. 91–94, 1997.

[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. London. A 317, pp. 319–340, 1970.

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

[DeR67] DeRosier, D. J., and A. Klug "Reconstruction of three-dimensional structures from electron micrographs", Nature, Vol. 217, pp. 130-134, 1967.

[DeR70] DeRosier, D. J., and P.B. Moore "Reconstruction of three-dimensional images from electron micrographs of structures with helical symmetry", J. Molec. Biol, Vol. 52, pp. 355-369, 1970

[Doe00] Doerschuk, P., and J. E. Johnson "Ab initio reconstruction and experimental design for cryo electron microscopy", IEEE Trans. on Information Theory, Vol. 46, No. 5, pp. 1714-1729, 2000.

[Dow91] K.H. Downing. "Spot-scan imaging in transmission electron microscopy", Science, 251, 53-59, 1991.

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

[Gon93] Gonzalez, R.C. and Woods, R.E. *Digital Image Processing*, Addison-Wesley Publishing Company, Inc, 1993.

[Gor74] Gordon, R., "Three-dimensional reconstruction from projections: A review of algorithms", Intern. Rev. of Cytology, Vol. 38, pp. 111–151, 1974.

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

[Har86] Harauz, G., and M. B. van Heel, "Exact filters for general geometry three dimensional reconstruction", Optik, Vol. 73, pp. 146-156, 1986.

[Heel00] van Heel, M., B. Growen, R. Matadeen, E. Orlova, R. Finn, T. Pape, D. Cohen, H. Stark, R. Schmidt, M. Schatz, and A. Patwardhan "Single-particle electron cryo-microscopy: towards atomic resolution", Quarterly Review of Biophysics, Vol. 33, No. 4, pp. 307-369, 2000.

[Her79] Herman, G. T., Editor: *Image reconstruction from projections, implementation and applications*, Springer-Verlag, 1979.

[Hhb81] Barrett, H. H., and W. Swindell, *Radialogical Imaging: The Theory of Image Formation, Detection and Processing*, New York, Academic Press, 1981.

[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", CS & E, pp. 555–559, 1994.

[KaS88] Kak, A. C., and M. Slaney, *Principles of Computerized Tomographic Imaging*, IEEE, Inc. New York, IEEE Press, 1988.

[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, 1997.

[Lyn99] Lynch, R. E., D. C. Marinescu, H. Lin, and T. S. Baker, "Parallel algorithms for 3D reconstruction of asymmetric objects from electron micrographs," Proc. IPPS/SPDP, IEEE Press, pp. 632–637, 1999.

[Lyn00] Lynch, R. E., H. Lin, and D. C. Marinescu, "An efficient algorithm for parallel 3D reconstruction of asymmetric objects from electron micrographs," Proc. Euro-Par 2000, Lecture Notes in Computer Science, Vol. 1900, pp. 481–490, 2000.

[Mar02] Marinescu, D. C. *Internet-Based Workflow Management: Towards a Semantic Web*, ISBN 0-471,43962-2, Wiley, 2002.

[MaM97] 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, Vol. 120, pp. 146–157, 1997.

[MaY01] Marinescu, D. C., Y. Ji, and R. E. Lynch "Space-Time Tradeoffs for Parallel 3D Reconstruction Algorithms for Virus Structure Determination", (Concurrency and Computation: Practice and Experience), 13:1083–1106, 2001.

[MaY02] Marinescu, D. C., Y. Ji, G. M. Marinescu "Grid Computing and Applications to Computational Biology", *Proc. Int. Symp.on Parallel and Distributed Computing*, Analele Stiintifice ale Universitatii Al.I.Cuza, Iasi, ISSN 1224-2268, vol XI, pp. 11-46, 2002.

[Mur97] Murshudov, G.N., A.A. Vagin, and E.J. Dodson, "Refinement of macromolecular structures by maximum likelihood method", Acta Cryst. D53, pp. 240-255, 1997

[Pan96] Pannu, N.S., and R.J. Read, "Improved structures refinement by maximum likelihood", Acta Cryst. D52, pp. 659-668, 1997

[Rad17] Radon, J., "Über the Bestimmung von Funktionen durch ihre Integralwerte langs gewisser Manningfaltigkeiten" Berichte über die Verhandlungen der Könieglich Sachsischen Gesellshaft der Wissenschaften zu Leiptzig. Math. Phys. Klasse, Vol. 69, pp. 262-277, 1917.

[Ram71] Ramanchandran, G. N., and A. V. Lakshminarayanan, "Three dimensional reconstructions from radiographs and electron micrographs: Application of convolution instead of Fourier transforms", Proc. National Academy of Sciences, Vol. 68, pp. 2236 - 2240, 1971.

[Ram95] Rao, R. P. V., R. D. Kriz, A. L. Abbott, and C. J. Ribbens, "Parallel implementation of the filtered back projection algorithms for tomographic imaging", http://www.sv.vt.edu/xray_ct/parallel/Parallel_CT.html

[Ros98] Rossmann, M. G., and Y. Tao, "Cryo-electron-microscopy reconstruction of partially symmetric objects", J. Structural Biology, Vol. 125, pp. 196–208, 1999.

[Yan00] Yan, X., N. H. Olson, J. L. Van Etten, M. Bergoin, M. G. Rossmann and T. S. Baker, "Structure and assembly of large, lipid-containing, dsDNA viruses", http://expert.cc.purdue.edu/~xdy/research/research.html.