Parallelization and Comparison of 3D Iterative Reconstruction Algorithms

J.R. Bilbao-Castro, J.M. Carazo
Centro Nacional de Biotecnología-CSIC
Biocomputing Unit
Campus Universidad Autónoma de Madrid.
Cantoblanco. 28049 MADRID
jrbcast@cnb.uam.es carazo@cnb.uam.es

J.J. Fernández, I. García
Universidad de Almería
Dept. Arquitectura Computadores y Electrónica
Universidad de Almería Ctra. Sacramento S/N,
La Cañada de San Urbano, 04120 ALMERIA
jose@ace.ual.es inma@ace.ual.es

Abstract

High resolution structure determination of biological macromolecules by electron microscopy is central to understand their biological function. These structural analyses involve processing thousands projection images taken from the specimen at different orientations. Regularized iterative reconstruction methods are well suited to deal with the extremely noise conditions found in those studies, but they are computationally expensive. Parallel computing then emerges as a natural solution for those problems allowing huge jobs to be run in clusters of workstations. This work describes and analyzes the parallel implementations of five 3D iterative reconstruction algorithms, including simultaneous and block-iterative methods. The evaluation of the parallel approaches is carried out in terms of speedups and computation versus communication times. It is shown that there are specific iterative methods that are specially well suited for parallelization, with a great level of scalability and fast convergence rates. This work draws the conclusion that the use of those parallel reconstruction methods is going to be central to afford "grand challenge" problems currently unapproachable in structural biology, such as structure determination at close-to-atomic resolution by electron microscopy.

1. Introduction

Electron microscopy (EM) is central to the study of many structural problems in BioSciences. Combined with image processing and three-dimensional (3D) reconstruction techniques, EM yields quantitative information about the 3D structure of biological specimens [7, 8], which is critical to understanding biological function at all levels of detail. This work focuses on the macromolecular domain (specimens with sizes around 200–100 Å, such as proteins).

Structural analyses of macromolecules involve processing thousands EM projection images taken from the specimen at very different orientations. In order to preserve as much detail as possible, the specimens are imaged at very low electron doses, which makes the EM images extremely noisy (SNR in the order of 0.1). The EM images are combined together through reconstruction algorithms to derive the 3D structure at enough resolution (typically in the range of 6–20 Å) to discern important structural features [7, 8].

Rigorous structural analyses require that image reconstruction introduces as little noise and artifact as possible at the spatial scales of interest, for a proper interpretation of the structure. Weighted back-projection (WBP) [16] is the standard method, whose relevance stems mainly from its computational simplicity. Its disadvantages, however, are (i) the sensitiveness to limited tilt angle conditions found in electron microscopy, and (ii) that it does not implicitly take into account the noise conditions nor the transfer function.

Series expansion reconstruction methods constitute one of the main alternatives to WBP to image reconstruction, and are getting increasing interest in this field [14, 6]. In general, these methods (i) yield smoother solutions under extremely noisy conditions and (ii) exhibit better behaviour under limited-angle conditions than WBP, which make them better suited for the problem of high-resolution structure determination of macromolecules. Despite their potential advantages [14], they still have not been extensively used due to their computational costs.

Series expansion methods can represent the density distribution in the volume by means of basis functions more general than the traditional voxels. During the nineties [13, 15, 14], overlapping spherically symmetric volume elements (*blobs*) with smooth transition to zero were thoroughly investigated as alternatives to voxels for image representation, concluding that blobs are better suited for representing natural structures. The use of blobs provides reconstruction algorithms with an implicit regularization mechanism, very appropriate for working under noisy conditions, yielding smoother reconstructions where artifacts and noise are reduced with relatively unimpaired resolution [6].

This work addresses the parallelization of blob-based series expansion methods in structure determination of macromolecules. It analyzes the use of traditional series expansion methods [12], such as Algebraic Reconstruction Techniques (ART), as well as other recently developed methods: such as Averaging Sequential Strings (ASS) [3] or Component Averaging methods (CAV) [4] characterized by a fast convergence. Parallel computing is used to face the huge computational requirements of all these methods, thus reducing the processing time. An analysis of the efficiency of the parallel approaches is carried out in terms of speedups and computation versus communication times.

2. Iterative Image Reconstruction Methods

2.1. Series Expansion Methods

Series expansion reconstruction methods assume that the 3D object or function f to be reconstructed can be approximated by a linear combination of a finite set of known and fixed basis functions b_j

$$f(r, \phi_1, \phi_2) \approx \sum_{j=1}^{J} x_j b_j(r, \phi_1, \phi_2) \quad (1)$$

(where (r, ϕ_1, ϕ_2) are spherical coordinates), and that the aim is to estimate the unknowns x_j. These methods also assume an image formation model where the measurements depend linearly on the object in such a way that:

$$y_i \approx \sum_{j=1}^{J} l_{i,j} x_j \quad (2)$$

where y_i denotes the i-th measurement of f and $l_{i,j}$ the value of the ith projection of the j-th basis function.

Under those assumptions, the image reconstruction problem can be modeled as the inverse problem of estimating the x_j's from the y_i's by solving the system of linear equations given by Eq. (2). Such systems of equations are typically solved by means of iterative methods.

2.2. ART and CAV Methods

ART methods constitute one of the best known families of iterative algorithms to solve such systems [12]. CAV methods have recently arisen [5, 4] as efficient iterative algorithms for solving large and sparse systems of linear equations. These methods have been derived from ART, with the important innovation of a weighting related to the sparsity of the system. This component-related weighting yields a convergence rate that may be far superior to the ART methods, specially at the early iteration steps.

Assuming that the whole set of equations in the linear system (Eq. (2)) may be subdivided into B blocks each of size S, a generalized version of iterative methods can be described via its iterative step from the k-th estimate to the $(k+1)$-th estimate by:

$$x_j^{k+1} = x_j^k + \lambda_k \sum_{s=1}^{S} \frac{y_i - \sum_{v=1}^{J} l_{i,v} x_v^k}{\sum_{v=1}^{J} w_v^b \, (l_{i,v})^2} l_{i,j} \quad (3)$$

where:

λ_k denotes the relaxation parameter.

$b = (k \bmod B)$, is the index of the block.

$i = bS + s$, is the i-th equation of the system.

w_v^b denotes the weighting factor.

The weighting factor is set up to S in the case of ART methods. In CAV methods, this factor w_v^b is set up to the number of times that the component x_v of the volume contributes with nonzero value to the equations in the b-th block.

The processing of all the equations in one of the blocks produces a new estimate. All blocks are processed in one iteration of the algorithm. This technique produces iterates which converge to a least squares solution of the system provided that the relaxation parameter is optimized [12, 4].

ART and CAV methods can be classified into the following three categories as a function of the number of blocks:

• **Sequential**. This version cycles through the equations one-by-one producing consecutive estimates ($S = 1$). This method exactly matches the well-known *row-action* ART method [12] and is characterized by a fast convergence as long as relaxation factors are optimized.

• **Simultaneous**. This version uses only one block ($B = 1$), considering all equations in the system in every iterative step. This version is inherently parallel, in the sense that every equation can be processed independently from the others. Simultaneous methods are characterized by better robustness under noise, but with a slow convergence rate. SIRT (Simultaneous Iterative Reconstruction Technique) [9] is the simultaneous version of ART. For CAV methods, that version is simply known as CAV [5].

• **Block-Iterative**. The Block-Iterative (BI) version represents the general case. In essence, this version sequentially cycles block-by-block, and every block is processed in a simultaneous way. BI methods (with $S > 1$) also exhibit an inherent nature for parallelization. The convergence rate is in-between of sequential and simultaneous versions, and is higher as block size decreases. SART (Simultaneous Algebraic Reconstruction Technique) [1] and BICAV [4] are the BI versions of ART and CAV methods, respectively.

This work focuses on the simultaneous and block iterative versions because of their inherent parallel nature. In this work, the blocks are chosen so that the number of equations is a multiple of the number of pixels in the EM projection images. In this way, all the equations involved by an EM image belong to the same block.

in the system carries out, essencially, the same task over its own data domain. The parallelization has been focused on distributed memory computers, in concrete clusters of workstations, using the message-passing paradigm.

In three-dimensional electron microscopy, the SPMD model consists of splitting the set of projection images into groups, and distributing the groups across the computing nodes. Fig. 2 shows a conceptual scheme of this approach. The nodes then process the images in parallel, and once per iteration, they communicate the results to obtain the solution at the current iteration. The data distribution and the processing of the groups depends on the iterative methods.

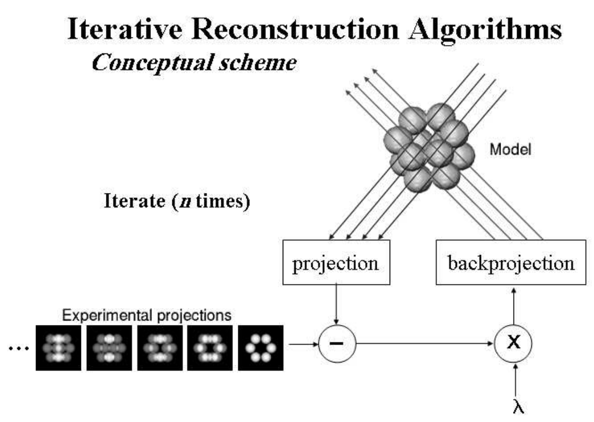

Figure 1. Conceptual scheme of iterative methods for 3D reconstruction from projections.

Conceptually, iterative methods (either CAV or ART) proceed in the following way (see Fig. 1): First, they start from an initial model, and the model is progressively refined as iterations evolve. For every iteration and every block: (1) the corresponding projections are computed from the current model; (2) the error between the experimental projections and those computed from the model is calculated; (3) the model is refined so that the error is minimized. In essence, this process is the one analytically expressed by Eq. (3).

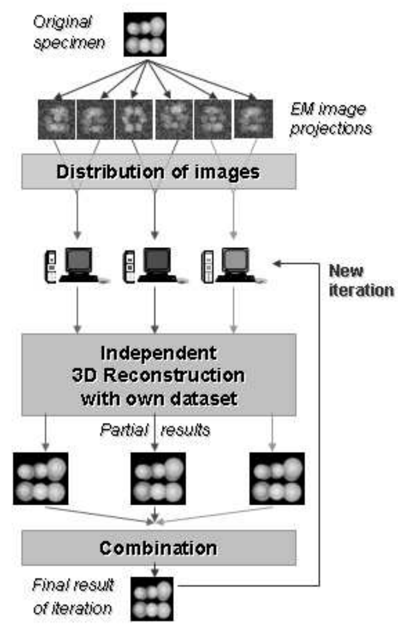

Figure 2. Conceptual Parallelization Scheme

2.3. Averaging Sequential Strings

Recently, another iterative algorithmic scheme to solve linear systems has arisen: Averaging Sequential Strings (ASS) [3]. This is also a block-iterative approach. The important difference with respect to the previous ones is that the equations in every block are processed sequentially, yielding a partial result, and the final result in the iteration is then computed as the weighted sum of the partial results. ASS is also inherently parallel in the sense that blocks can be processed independently. The convergence rate is inbetween of sequential and simultaneous versions of ART and CAV.

2.4. Parallelization

In this work, the parallel strategies that have been developed are based on *domain decomposition* and the SPMD model (*Single Program, Multiple Data*) in which every node

The parallelization of the simultaneous and BI versions of ART/CAV then consists of distributing the EM images of the current block of equations into the computing nodes. Every node then computes its partial result following a simultaneous scheme with its equations. Finally, the nodes communicate each other to yield the final result through a weigthed sum of the partial results. Then, a new block is processed and so forth. Therefore, simultaneous algorithms require only a global reduction operation per iteration, whereas BI methods require a number of communications per iteration which is inversely proportional to the block size. With this parallelization approach, parallel simultaneous and BI methods are guaranteed to yield the same results as the corresponding sequential version.

In other words, the convergence rate of the method is preserved.

The parallelization of ASS follows a different approach. In each iteration, the blocks of equations are distributed across the nodes, i.e. each node receives a block of equations. Every node then computes its partial result following a sequential or a BI scheme over its equations. Finally, the nodes communicate each other to yield the final result by a global reduction operation. Conceptually, this parallelization approach is equivalent to "independent" reconstructions computed by the different nodes and they are combined together in every iteration. This strategy proves to be interesting to analyze from the mathematical point of view due to the implications in the convergence rate of the algorithm. Parallelization allows to reduce the computation time per iteration but, however, may reduce the convergence rate of the algorithm. It has been formulated a heuristic rule over the relaxation parameter of the algorithm so that the sequential convergence rate is maintained [2]:

$$\lambda_{N_p} = \lambda_{\text{Seq}} \times N_p$$

where N_p denotes the number of processors, λ_{Seq} is the relaxation parameter used in the sequential version and λ_{N_p} is the value that should be used in the parallel version to get similar convergence rate.

3. Results

This section is intended to show the efficiency of the parallelization approaches above described in their application to three-dimensional electron microscopy. Speedups, computation and communication times have been computed for the different reconstruction methods parallelized here.

A model of Bacteriorhodopsin, the first macromolecule solved at atomic resolution by electron microscopy [11] has been used to perform the reconstructions. A set of 1000 projections, randomly distributed in projection space and contaminated with Gaussian noise at SNR=1/3, was calculated from the model. Then, the set of projections was used to compute the reconstructions by following the different parallel iterative methods. Fifty iterations of each method were used to calculate speedups and times. The reconstructions and the projections were $64 \times 64 \times 64$ voxels and 64×64 pixels in size, respectively.

Five different reconstruction algorithms have been tested: ASS, SIRT, CAV, SART and BICAV. For BI algorithms (SART and BICAV), two different block sizes have been tested: 16 and 48 projections (called in this paper SART-16, SART-48, BICAV-16 and BICAV-48, respectively). That way, the impact of block sizes on the parallelization could be measured.

A 8-nodes (uniprocessor) Linux cluster of workstations has been used to run the experiments. The characteristics of this cluster are:

- 8 nodes: PIII 933 Mhz. 512 MB RAM, 20 GB HDD, 2 x 1Gb Ethernet NICs. Cross-mounted /home filesystem.
- 1 Server Node: 2x 2.66 Ghz PIII Xeon. 2 x 73 GB SCSI-320. 2 GB RAM. 2 x 1 Gb Ethernet NICs.

MPICH 1.2.5 was installed as Message-Passing Interface (MPI) library [10]. To assure exclusive access to nodes so times cannot be altered by other users, a batch scheduling policy has been implemented on both server and nodes. Therefore, jobs remain queued until requested resources are available so they can be assigned to it in exclusive mode. OpenPBS was used as the batch scheduling system.

3.1. Speedups and execution times

Speedup is the most often used measure of parallel performance, and was computed in this work to assess the efficiency of the parallel approaches. Speedup is defined as the ratio between the turnaround times in the sequential and in the parallel version of the program. Table 3.1 represents the speedups obtained for the different algorithms tested and using up to 8 processors.

	2 p.	3 p.	4 p.	5 p.	6 p.	7 p.	8 p.
ASS	2,0	2,98	3,88	4,88	5,8	6,72	7,63
CAV	2,0	2,98	3,92	4,94	5,85	6,75	7,67
SIRT	2,0	3	3,89	4,89	5,82	6,71	7,49
BICAV-48	1,91	2,73	3,56	4,11	4,86	5,25	6,19
SART-48	1,92	2,73	3,57	4,01	4,77	5,22	5,97
BICAV-16	1,83	2,42	3,12	3,14	3,3	3,75	4,54
SART-16	1,85	2,37	3,08	3,07	3,21	3,72	4,36

Table 1. Speedups for the different reconstruction methods and number of processors

Fig. 3 shows a graph with the speedup curves for the different methods versus the number of processors. It is clearly observed that ASS, and the simultaneous algorithms, SIRT and CAV, exhibit the best speedup curves with a nearly linear behaviour. However, the BI methods SART and BICAV exhibit poorer speedups, and the behaviour is worse as the block size decreases. This is justified by the fact that BI algorithms require a number of global reduction operations per iterations proportional to the number of blocks. In order to quantify the cost of these operations, the following section presents the time devoted to communications for the different algorithms.

Both Fig. 4 and Table 2 show the average execution time (including computation and communication) per iteration for the different tested algorithms. This figure is intended to compare the computation costs of the different algorithms. It is clearly seen that ASS, SIRT and CAV require less time per iteration. However, BI algorithms are more expensive, and their demands depend inversely on the block

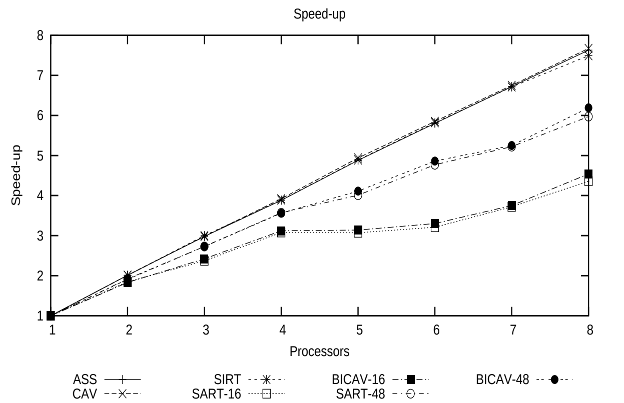

Figure 3. Speedup vs no. processors

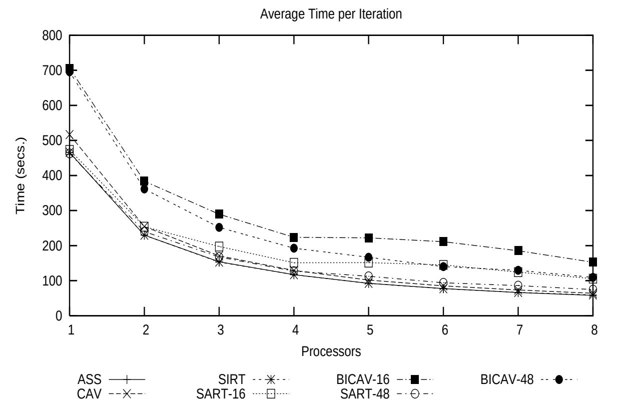

Figure 4. Average execution time per iteration

size. Fig. 4 shows that BICAV algorithms are significantly more demanding than SART due to the computation and transmission of the weights for each block.

	1 p.	2 p.	3 p.	4 p.	5 p.	6 p.	7 p.	8 p.
ASS	465.7	229.9	153.9	117.3	92.7	77.46	66.3	58
CAV	516.6	254.7	170.8	129.3	101.9	85.44	73.6	64.3
SIRT	466.3	229.9	153	117.2	92.6	77.3	66.6	59.2
BICAV-48	695.6	361.7	252.3	192.6	166.8	140.3	129.7	109.4
SART-48	463.2	240.1	167.5	127.3	112.8	94.2	85.9	74.6
BICAV-16	705.9	383.8	289.5	223.8	222.3	211.2	185.46	152.7
SART-16	473.1	254.1	197.4	151	151.7	145	124.52	105.8

Table 2. Average time per iteration vs. number of processors.

	2 p.	3 p.	4 p.	5 p.	6 p.	7 p.	8 p.
ASS	0	0	0	0,01	0,01	0,01	0,01
CAV	0	0	0	0,01	0,01	0,01	0,01
SIRT	0	0	0	0,01	0,01	0,01	0,01
BICAV-48	0,03	0,06	0,06	0,14	0,14	0,19	0,16
SART-48	0,02	0,06	0,06	0,15	0,14	0,16	0,16
BICAV-16	0,07	0,15	0,16	0,3	0,36	0,36	0,33
SART-16	0,05	0,15	0,15	0,29	0,34	0,33	0,32

Table 3. Ratio communication time vs total execution times, shown for the different reconstruction methods and number of processors

3.2. Communication times

Communication is usually one of the factors which influences the degree of parallelism that an application can reach (scalability). In this section this parameter is analyzed in order to discern which algorithms are more suitable for parallelization and which ones scale badly.

Table 3 represents the ratio between the communication time and the turnaround time for the different reconstruction methods and number of processors. Fig. 5 shows a graph with the curves corresponding to the data in Table 3.

As expected, ASS, CAV and SIRT algorithms are the least affected by the communication. They perform just a communication per iteration to share the partial results and, consequently, the communication time is negligible (less than 1%) compared to the time dedicated to computation. Therefore, these algorithms are not strongly affected by communications and they can thus achieve good scalability.

However, a different behaviour is found in BI algorithms (BICAV and SART) as they perform more communications as block size decreases. They may be severely affected by communications, and this effect is stronger as the number of processors increases. Table 3 show that more than 30% of the total execution time is dedicated to communication in SART and BICAV when block size is 48 and more than 4 processors are used. This behaviour makes those BI algorithms less suitable for parallelization.

There has to be remarked that the communications in the reconstruction algorithms follow the collective pattern, which means that all the processes synchronize at a point and share all-to-all their information. As MPI is being used, the "MPI_ALLREDUCE" operation, which is optimized for this kind of operations, has been chosen. It adds the possibility to perform a collective operation over the shared data like a sum, division, average and others.

We can observe in Fig. 5 that curves behave better for numbers of processors which are power of two. However, for other numbers of processors the time for collective communications is high. This can be explained because a "MPI_ALLREDUCE" uses tree structures as data distribution patterns. Number of processors that are not powers of

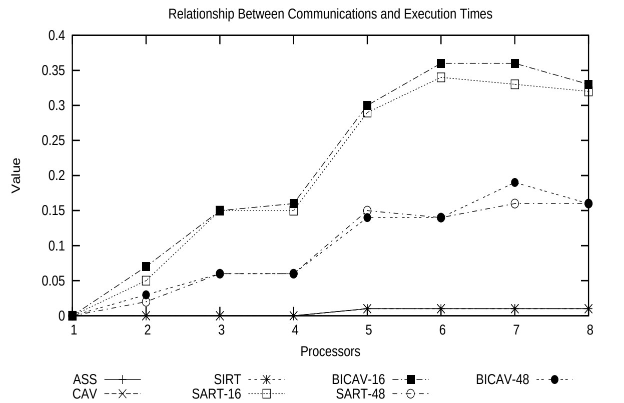

Figure 5. Ratio Comms. Time/Total Time vs n. processors

two would involve unbalanced trees, which may justify this poor behaviour.

4. Discussion and Conclusion

This work has described and analyzed parallel approaches for regularized iterative reconstruction methods used in three-dimensional electron microscopy. Several types of algorithms have been parallelized and analyzed: simultaneous methods (SIRT, CAV) and block-iterative methods (SART, BICAV, ASS).

The results that have been obtained show that ASS, SIRT and CAV exhibit similar behaviour in terms of scalability, showing a nearly linear speedup. Therefore, all of them are well suited for parallelization. However, block-iterative algorithm exhibit a poor behaviour in terms of scalability as the block size is reduced. This is a strong disadvantage as little block sizes are desirable in order to get fast convergence rates in reconstruction quality [12, 2].

A previous work [2] showed that ASS exhibits the highest convergence rate in terms of quality of the reconstructions. The present work has shown that ASS is one of the methods with higher scalability. Therefore, the conjunction of these two facts makes ASS the best method for large-scale three-dimensional reconstruction from projections.

Therefore, parallel versions of Iterative Reconstruction Algorithms –ASS in particular– have shown to be effective enough to substitute sequential versions without loss of convergence. Previously unaffordable large reconstructions could then be faced and carried out now in a cluster of workstations in a relatively short computation time.

The use of parallel computing is becoming essential to afford "grand challenge" problems currently unapproachable in structural biology. High resolution structure determination of biological specimens by electron microscopy is currently trying to reach close-to-atomic resolution by combining hundreds of thousands or even one million images of the specimen under study [8]. The computational costs that this challenge involves make parallel computing central in this field. This work has shown that parallel and distributed computing can be effectively used for image reconstruction, and proposes ASS as the method of choice.

5. Acknowledgements

This work has been partially supported by the Spanish CICYT through grants TIC2002-00228 and BIO2001-1237 and The European Commission through grant HPRI-CT-1999-00026 (the TRACS Programme at EPCC).

References

[1] A. H. Andersen and A. C. Kak. Simultaneous algebraic reconstruction technique (SART): A superior implementation of the art algorithm. *Ultrasonic Imaging*, 6:81–94, 1984.

[2] J. R. Bilbao-Castro, J. M. Carazo, I. García, and J. J. Fernández. Parallel iterative reconstruction methods for structure determination of biological specimens by electron microscopy. In *Proceedings of IEEE International Congress on Image Processing (ICIP)*, pages 565–568, 2003.

[3] Y. Censor, T. Elfving, and G. T. Herman. *Inherently Parallel Algorithms in Feasibility and Optimization and Their Applications*, chapter Averaging Strings of Sequential Iterations for Convex Feasibility Problems, pages 101–114. Elsevier Science, 2001.

[4] Y. Censor, D. Gordon, and R. Gordon. BICAV: A block-iterative, parallel algorithm for sparse systems with pixel-related weighting. *IEEE Trans. Med. Imag.*, 20:1050–1060, 2001.

[5] Y. Censor, D. Gordon, and R. Gordon. Component averaging: An efficient iterative parallel algorithm for large and sparse unstructured problems. *Parallel Computing*, 27:777–808, 2001.

[6] J. J. Fernández, A. F. Lawrence, J. Roca, I. García, M. H. Ellisman, and J. M. Carazo. High performance electron tomography of complex biological specimens. *J. Struct. Biol.*, 138:6–20, 2002.

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

[8] J. Frank. Single-particle imaging of macromolecules by cryo-electron microscopy. *Annu. Rev. Biophys. Biomol. Struct.*, 31:303–319, 2002.

[9] P. Gilbert. Iterative methods for the 3D reconstruction of an object from projections. *J. Theor. Biol.*, 76:105–117, 1972.

[10] W. Gropp, E. Lusk, and A. Skjellum. *Using MPI Portable Parallel Programming with the Message-Passing Interface*. MIT Press, 1994.

[11] R. Henderson, J. M. Baldwin, T. A. Ceska, F. Zemlin, E. Beckmann, and K. H. Downing. Model for the structure of bacteriorhodopsin based on high-resolution electron cryo-microscopy. *J. Mol. Biol.*, 213:899–929, 1990.

[12] G. T. Herman. *Medical Imaging. Systems, Techniques and Applications. C. Leondes (Ed.)*, chapter Algebraic Reconstruction Techniques in Medical Imaging, pages 1–42. Gordon and Breach Science, 1998.

[13] R. M. Lewitt. Alternatives to voxels for image representation in iterative reconstruction algorithms. *Phys. Med. Biol.*, 37:705–716, 1992.

[14] R. Marabini, G. T. Herman, and J. M. Carazo. 3D reconstruction in electron microscopy using ART with smooth spherically symmetric volume elements (blobs). *Ultramicroscopy*, 72:53–56, 1998.

[15] S. Matej, R. M. Lewitt, and G. T. Herman. Practical considerations for 3D image reconstruction using spherically symmetric volume elements. *IEEE Trans. Med. Imag.*, 15:68–78, 1996.

[16] M. Rademacher. *Electron Tomography, J. Frank (Ed.)*, chapter Weighted Back-Projection Methods, pages 91–115. Plenum Press, 1992.