High-performance electron tomography of complex biological specimens

José-Jesús Fernández, a Albert F. Lawrence, b Javier Roca, a Inmaculada García, a
Mark H. Ellisman, b and José-María Carazo c,*

a Departamento de Arquitectura de Computadores, Universidad de Almería, 04120 Almería, Spain
b National Center for Microscopy and Imaging, University of California, San Diego, La Jolla, CA 92093-0608, USA
c Biocomputing Unit, Centro Nacional de Biotecnología, Universidad Autónoma, 28049 Madrid, Spain

Received 10 December 2001; and in revised form 15 February 2002

Abstract

We have evaluated reconstruction methods using smooth basis functions in the electron tomography of complex biological specimens. In particular, we have investigated series expansion methods, with special emphasis on parallel computation. Among the methods investigated, the component averaging techniques have proven to be most efficient and have generally shown fast convergence rates. The use of smooth basis functions provides the reconstruction algorithms with an implicit regularization mechanism, very appropriate for noisy conditions. Furthermore, we have applied high-performance computing (HPC) techniques to address the computational requirements demanded by the reconstruction of large volumes. One of the standard techniques in parallel computing, domain decomposition, has yielded an effective computational algorithm which hides the latencies due to interprocessor communication. We present comparisons with weighted back-projection (WBP), one of the standard reconstruction methods in the areas of computational demand and reconstruction quality under noisy conditions. These techniques yield better results, according to objective measures of quality, than the weighted backprojection techniques after a very few iterations. As a consequence, the combination of efficient iterative algorithms and HPC techniques has proven to be well suited to the reconstruction of large biological specimens in electron tomography, yielding solutions in reasonable computation times. © 2002 Elsevier Science (USA). All rights reserved.

Keywords: Electron tomography; High-performance computing; Iterative reconstruction algorithms; Parallel computing

1. Introduction

Electron microscopy is central to the study of many structural problems in modern biology, biotechnology, biomedicine, and other related fields. Electron microscopy together with sophisticated image processing and three-dimensional (3D) reconstruction techniques yields quantitative structural information about the 3D structure of biological specimens (Frank, 1992; Koster et al., 1997; McEwen and Marko, 2001; Perkins et al., 1997). Knowledge of three-dimensional structure is critical to understanding biological function at all levels of detail.

In contrast to earlier instruments, high-voltage electron microscopes (HVEMs) are able to image relatively thick specimens that contain complex 3D structure.

Electron tomography simplifies determination of complex 3D structures and their subsequent analysis. This method requires a set of HVEM images acquired at different orientations, via tilting the specimen around one or more axes (Mastronarde, 1997; Penczek et al., 1995; Perkins et al., 1997). Rigorous structural analyses require that image acquisition and reconstruction introduce as little noise and artifact as possible at the spatial scales of interest, for a proper interpretation and measurement of the structural features. As a consequence of the need for structural information over a relatively wide range of spatial scales, electron tomography of complex biological specimens requires large projection HVEM images (typically 1024 × 1024 pixels or larger). Electron tomography on this scale yields large reconstruction files and requires an extensive use of computational resources and considerable processing time.

*Corresponding author. Fax: +34-915-854-506.
E-mail address: carazo@cnb.uam.es (J.-M. Carazo).
High-performance computing (HPC) addresses such computational requirements. HPC involves the use of: (i) parallel computing on supercomputers or networks of workstations, (ii) sophisticated code optimization techniques, (iii) intelligent use of the hierarchical memory systems in the computers, and (iv) awareness of communication latencies.

Weighted back-projection (WBP) (Rademacher, 1992) has been one of the most popular methods in the field of electron tomography of large specimens. The relevance of WBP in this field stems from: (i) the computational simplicity of the method and (ii) the fact that the method is linear, and therefore the outcome is predictable in a straightforward way by the experimental tomographic data. The main disadvantages of WBP are that (i) the results may be strongly affected by limited tilt angle data due to the limited tilt capabilities of the HVEM and (ii) WBP does not implicitly take into account the noise conditions or the response to the HVEM. As a consequence of the latter, a posteriori regularization techniques (such as low-pass filtering) may be needed to attenuate the effects of the noise.

Series expansion reconstruction methods constitute one of the main alternatives to WBP to image reconstruction. Despite their potential advantages (Marabini et al., 1997, 1998, 1999), these methods still have not been extensively used in the field of electron tomography due to their high computational costs. One of the main advantages of these methods is the capability of representing the density function in the volume of the distribution by means of basis functions more general than the traditional voxels. During the 1990s (Lewitt, 1990, 1992; Matej and Lewitt, 1995; Matej et al., 1996b), spherically symmetric volume elements (blobs) have been thoroughly investigated and, as a consequence, the conclusion that blobs yield better reconstructions than voxels has been drawn in the fields of medicine (Furuie et al., 1994; Kinahan et al., 1995; Matej et al., 1994, 1996a; Obi et al., 2000), three-dimensional electron microscopy (Marabini et al., 1998, 1999) and even electron tomography (Marabini et al., 1997).

The work addresses the application of series expansion methods using blobs as basis functions in electron tomography of complex biological specimens from the perspective of computational speed. This work explores the use of recently developed series expansion methods (Censor et al., 2001a,b) (component averaging methods) characterized by a really fast convergence, which achieve least-squares solutions in a very few iterations. Parallelization and other HPC techniques yield determinations of the 3D structure of large volumes in reasonable computation time. In addition, the use of blob basis functions provides the series expansion methods with an implicit regularization mechanism which makes them better suited to noisy conditions than WBP. The aim of this work is to show that blob-based series expansion methods may constitute real alternatives to WBP as far as the trade-off of computation time vs quality in the reconstruction is concerned.

In the following, Section 2 is devoted to the iterative reconstruction methods and provides a review of the main concepts and a description of the component averaging methods. The concept of blobs and their main properties are also briefly described. Section 3 is focused on the HPC approach. Section 4 presents the experimental results that have been obtained. We discuss computational times and speed-up rates resulting from the parallelization as well as the measures of quality provided by the reconstruction algorithms, iteration by iteration. Finally, the results are analyzed in detail in Section 5, proposing guidelines on the use of blob-based component averaging methods in electron tomography.

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 \( b_j \)

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

where \( (r, \phi_1, \phi_2) \) are spherical coordinates, and that the aim is to estimate the unknowns \( x_j \). These methods are based on an image formation model where the measurements depend linearly on the object in such a way that

\[
y_i \approx \sum_{j=1}^{J} l_{ij} x_j,
\]

where \( y_i \) denotes the \( i \)th measurement of \( f \) and \( l_{ij} \) the value of the \( i \)th 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). Algebraic reconstruction techniques (ART) constitute one of the best known families of iterative algorithms to solve such systems (Herman, 1998).

Series expansion methods have some advantages over weighted back-projection: first, in the flexibility in the spatial relationships between the object to be reconstructed and the measurements taken, and second, in the possibility of incorporating spatial constraints. The former involves the possibility of using different basis functions, grids to arrange them, and spacing among them. The latter implies that any a priori information about the object or the image formation process may be used to control the solution.
The main drawbacks of series expansion methods are (i) their high computational demands and (ii) the need for parameter optimization to properly tune the model (basis functions, the image formation, estimation criterion). The high computational requirements may be reduced by using the efficient iterative methods that will be described below. Also, the proper selection of parameters may speed computations (Herman, 1998).

2.2. Component averaging methods

Component averaging methods have been developed recently (Censor et al., 2001a,b) as efficient iterative algorithms for solving large and sparse systems of linear equations. In essence, these methods have been derived from ART methods (Herman, 1998), with the important innovation of a weighting related to the sparsity of the system. This component-related weighting provides the methods with a convergence rate that may be far superior to the ART methods. 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 component averaging methods can be described via its iterative step from the $k$th estimate to the $(k+1)$th estimate by

$$x_{i}^{k+1} = x_{i}^{k} + \lambda_{k} \sum_{s=1}^{S} \left( y_{i} - \sum_{j=1}^{J} I_{i,j}x_{j}^{k} \right) \frac{I_{i,s}}{\sum_{s'=1}^{S} I_{i,s'}}$$

where $\lambda_{k}$ denotes the relaxation parameter; $b = (k \mod B)$ and denotes the index of the block; $i = bS + s$ and represents the $i$th equation of the whole system; and $s_{v}$ denotes the number of times that the component $x_{i}$ 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 weighted least-squares solution of the system of equations provided that the relaxation parameters are within a certain range and the system is consistent (Censor et al., 2001a).

The efficiency of component averaging methods stems from the explicit use of the sparsity of the system, represented by the $s_{v}$ term in Eq. (3). The component-related weighting makes component averaging methods progress through the iterations based on oblique projections onto the hyperplanes constituting the linear system (Censor et al., 2001a,b). Traditional ART methods, on the other hand, are based on orthogonal projections (in ART methods, $s_{v} = 1$). Oblique projections allow component averaging methods to have a fast convergence rate, especially at the early iterate steps.

Component averaging methods, like ART methods (Censor and Zenios, 1997), can be classified into the following categories as a function of the number of blocks involved.

2.2.1. Sequential

In this version, every equation in the system constitutes a block by itself ($S = 1$). As a consequence, the method cycles through the equations one by one, producing consecutive estimates. This version exactly matches the well-known row-action ART method since, in such a case, the weighting is useless. These methods are characterized by a fast convergence as long as (typically) small relaxation factors are chosen for the inconsistent case. These methods were originated with Kaczmarz (1937).

2.2.2. Simultaneous

This version is known simply as “component averaging” (CAV) (Censor et al., 2001b). In these methods the number of blocks is $B = 1$, and, consequently, all of the equations in the system are considered in every iterative step. CAV takes into account the number of times each component $x_{i}$ contributes in the whole set of equations in the system. These methods are inherently parallel, in the sense that every equation can be processed independently of the others. Simultaneous methods are characterized, in general, by a slow convergence rate. However, CAV has an initial convergence rate far superior to simultaneous ART methods (SIRT, for instance). Furthermore, the convergence of this method has been proven in the consistent and inconsistent cases (Censor et al., 2001b).

2.2.3. Block-iterative

These represent the general case. The block-iterative version of CAV has been named BICAV (Censor et al., 2001a) and represents the general case of these methods whose analytical form is given by Eq. (3). In essence, these methods sequentially cycle block by block, and every block is processed in a simultaneous way. As far as the weighting is concerned, BICAV acts like CAV but computes the weighting block by block. BICAV has an initial convergence rate significantly superior to that of CAV and on a par with row-action ART methods, provided that the block size and relaxation parameter are optimized. The convergence of BICAV has been proven in the consistent case (Censor et al., 2001a). BICAV (with $S > 1$) is also inherently parallel in the sense that all the equations in every block may be processed in parallel.

2.3. Basis functions

In the field of image reconstruction, the choice of the set of basis functions to represent the object to be reconstructed greatly influences the result of the algorithm (Lewitt, 1992). Spherically symmetric volume elements (blobs) with smooth transition to zero have been thoroughly investigated (Lewitt, 1990, 1992; Matej and Lewitt, 1995, 1996b) as alternatives to voxels for image representation, concluding that the properties of blobs make them well suited to represent natural structures of
all physical sizes. The use of blob basis functions provides the reconstruction algorithm with an implicit regularization mechanism. In that sense, blobs are especially suited to work under noisy conditions, yielding smoother reconstructions where artifacts and noise are reduced with relatively unimpaired resolution. Specifically in electron tomography, the potential, for a number of specific tasks, of blob-based iterative reconstruction algorithms with respect to WBP was already highlighted by means of an objective task-oriented comparison methodology (Marabini et al., 1997).

Blobs are a generalization of a well-known class of window functions in digital signal processing called Kaiser–Bessel (Lewitt, 1990). Blobs are spatially limited and, in practice, can also be considered band-limited because of the analytical restrictions on the density and its derivatives. The shape of the blob and its spectral features are controlled by three parameters: the radius \( a \) of the blob, the differentiability order \( m \) (normally set up to 2), and the density drop-off \( z \). Their appropriate selection is highly important. The blob full-width at half-maximum (FWHM), determined by the parameters, is chosen on the basis of a compromise between the resolution desired and the data noise suppression: narrower blobs provide better resolution, and wider blobs allow better noise suppression. Further, the parameters are chosen in such a way that the frequency spectrum of the blob has a value of zero at the sampling frequency, for anti-aliasing purposes. For a detailed description of blobs, refer to Lewitt (1992).

The basis functions in Eq. (1) are shifted versions of the selected blob. The arrangement of the center of those shifted versions is referred to as a grid. As far as blobs are concerned, there are several options to select a grid (Matej and Lewitt, 1995). In this work, we have used the same grid as used with voxels: the simple cubic grid, with the important difference that blobs overlap. The arrangement of overlapping blobs covers the space with a pseudo-continuous density distribution (see Fig. 1) very suitable for image representation.

The use of blobs allows an efficient computation of the forward- and backward-projection stages in any iterative, either ART or component averaging, method. The spherical symmetry of blobs makes the projection of the blob along any direction the same. Consequently, it is possible to precompute the projection of the generic blob and store it in a look-up table (Matej et al., 1996b). In this way, the computations of the \( l_{ij} \) terms (Eq. (3)) in the forward- and backward-projection stages are transformed into simple references to the look-up table. Furthermore, a blob-driven approach allows the use of incremental algorithms. Consequently, the use of blobs as basis functions enables substantial computational savings compared to the use of voxels.

3. High-performance computing in electron tomography

Parallel computing has been widely investigated for many years as a means of providing HPC facilities for
large-scale and grand-challenge applications. In the field of electron tomography of large specimens, parallel computing has already been applied to large-scale reconstructions by means of WBP as well as voxel-based iterative methods (Perkins et al., 1997). The use of voxels as basis functions makes the single-tilt axis reconstruction algorithms relatively straightforward to implement on massively parallel supercomputers. The reconstruction of each of the one-voxel-thick slices orthogonal to the tilt axis of the volume is assigned to an individual node on the parallel computer. In that sense, this is an example of an embarrassingly parallel restructuring of the reconstruction problem (see Wilkinson and Allen (1999) for a review of the most important concepts in parallel computing) that is no longer valid for blob-based methodologies, as will be shown below. For the blob case, more effort is needed for a proper data decomposition and distribution across the nodes.

3.1. Data decomposition

The single-tilt axis data acquisition geometries typically used in electron tomography allow the application of the Single-Program Multiple-Data (SPMD) computational model for parallel computing. In the SPMD model, all the nodes in the parallel computer execute the same program, but for a data subdomain. Single-tilt axis geometries allow a data decomposition into slabs of slices orthogonal to the tilt axis. In the SPMD model for this decomposition, the number of slabs equals the number of nodes, and each node reconstructs its own slab.

Those slabs of slices would be independent if voxel basis functions were used. However, the use of overlapping blobs as basis functions makes the slices, and consequently the slabs, interdependent (see Fig. 1). Therefore, each of the nodes in the parallel computer receives a slab composed of its corresponding subdomain together with additional redundant slices from the neighbor nodes. The number of redundant slices depends on the blob extension. Fig. 2a shows the scheme of the data decomposition.

The slices in the slab received by a given node are classified into the following categories (see scheme in Fig. 2b):

**Halo.** These slices are only needed by the node to reconstruct some of its own slices. They are the redundant slices mentioned above and are located at the extremes of the slab. Halo slices come from neighbor nodes, where they are reconstructed.

**Unique.** These slices are to be reconstructed by the node. In the reconstruction process, information from neighbor slices is used. These slices are further divided into the following subcategories:

**Edge.** Edge slices are slices that require information from the halo slices originating from the neighbor node.

**Own.** Own slices are slices that do not require any information from halo slices. As a result, these slices are independent of those in the neighbor nodes.

It should be noted that edge slices in a slab are halo slices in a neighbor slab.

![Fig. 2. Data decomposition.](image-url)

(a) The slices orthogonal to the tilt axis are decomposed into slabs of slices. The number of slabs equals the number of nodes. Every node in the parallel system receives a slab. The slab includes a set of slices that are unique (shaded in light gray) and additional redundant slices (shaded by dark gray) according to the blob extension. (b) Classification of the slices in the slab. Halo slices (in dark gray) are those redundant slices originate from neighbor nodes. Unique slices are slices to be reconstructed by the node and are subdivided into edge and own slices. Edge slices (in medium gray) are those unique slices which require information provided by halo slices, whereas own slices (light gray) do not.
3.2. The parallel iterative reconstruction method

In this work, the block-iterative (with $S > 1$) and the simultaneous versions of the component averaging methods have been parallelized following the SPMD model and the data decomposition just described. The row-action version of those methods has been discarded since the BICAV method yields a convergence rate roughly equivalent to it but with better speed-up due to its inherently parallel nature.

Conceptually, block-iterative and simultaneous reconstruction algorithms may be decomposed into three subsequent stages: (i) computation of the forward-projection of the model; (ii) computation of the error between the experimental and the calculated projections; and (iii) refinement of the model by means of back-projection of the error. Those stages can be easily identified in Eq. (3). Those reconstruction algorithms pass through those stages for every block of equations and for every iteration, as sketched in Fig. 3a. Initially, the model may be set up to zero, a constant value, or even the result of another reconstruction method.

The interdependence among neighbor slices due to the blob extension implies that, in order to compute either the forward-projection or the error back-projection for a given slab, there must be a proper exchange of information between neighbor nodes. Specifically, updated halo slices are required for a correct forward-projection of the edge slices. On the other hand, updated halo error differences are needed for a proper error back-projection of the edge slices. The need for communication between neighbor nodes for a mutual update of halo slices therefore arises. The flow chart in Fig. 3a shows a scheme of the iterative algorithm, indicating the communication points just before and after the error back-projection. Fig. 3b also shows another scheme depicting the slices involved in the communications: halo slices are updated with edge slices from the neighbor node.

Our parallel SPMD approach then allows each of the nodes in the parallel computer to independently process the data with information from edge data in the neighbor nodes. In the communication point just before the error back-projection step, the data to be exchanged are the error differences. However, after the error back-projection, the reconstructed slices are the data exchanged.

Fig. 3. The parallel iterative reconstruction method. (a) Flow chart of the iterative reconstruction algorithm. Once the volume is initiated, the three main stages in the iterative method are (i) forward-projection, (ii) error computation, and (iii) model refinement by means of error back-projection. These stages are iteratively passed through for every new block of equations to be processed and for every new iteration of the algorithm. In the parallel version, there are two communication points per iterative step in the algorithm so as to exchange (i) information on the error as well as (ii) reconstructed slices. (b) Communications in the parallel algorithm. In the two communication points of the algorithm, every node updates its halo data with information from edge data in the neighbor nodes. In the communication point just before the error back-projection step, the data to be exchanged are the error differences. However, after the error back-projection, the reconstructed slices are the data exchanged.
its own slab of slices. This fact notwithstanding, there are two implicit synchronization points in every pass of the algorithm in which the nodes must wait for the neighbors. Those implicit synchronization points are the communication points just described.

In any parallelization project where communication between nodes is involved, latency hiding becomes an issue. That term stands for overlapping communication and computation so as to keep the processor busy while waiting for the communications to be completed. In this work, an approach that further exploits the data decomposition has been devised for latency hiding. In essence, the approach is based on ordering the way the slices are processed between communication points. Fig. 4 sketches this approach. First of all, the left edge slices are processed, and they are sent as soon as they are ready. The communication of left edge slices is then overlapped with the processing of right edge slices. Similarly, the communication of right edge slices is overlapped with the processing of own slices. This strategy is applied to both communication points, just before and after the error back-projection stage. On the other hand, the ordered way the nodes communicate with one another also makes this parallel approach deadlock-free.

Finally, the data decomposition described in the previous subsection makes our parallel SPMD approach implicitly load balanced since all the nodes receive a slab of the same (or as similar as possible) size. This load balancing capability holds as long as the parallel system where the application is to be executed is homogeneous in workload (for example, in dedicated systems). Nonetheless, the parallel system might be overwhelmed in some situations due to the workload. Such situations might arise if the slabs were so huge that they did not fit into the memory of the nodes.

3.3. Cluster computing

The availability of high-speed networks and increasingly powerful commodity microprocessors are making clusters of workstations a readily available alternative to expensive, large and specialized HPC platforms. Cluster computing (Buyya, 1999) turns out to be a cost-effective vehicle for supercomputing, based on the usage of commodity hardware and standard software components, an increasingly popular alternative. Furthermore, clusters of workstations have the important advantage over supercomputers in that the turnaround time (the time elapsed from the program launching until the results are available) is much lower. This is due to the usually long wait times in the queues of supercomputers.

The availability of application programming interfaces (APIs) such as message-passing interface (MPI) (Gropp et al., 1994) allows programmers to implement

---

**Fig. 4. Latency hiding: Overlapping communication and computation.** The approach that has been devised for latency hiding is based on the explicit ordering in processing the slices within a slab. Processing the slices in the order (i) left edge, (ii) right edge, and (iii) own, allows us to overlap communication with computation as shown in this figure. The boxes represent stages of the iterative algorithm, and the discontinuous lines denote the transmission of the data already processed. The latency hiding is applied for both communication points in the parallel algorithm: just before and after the error back-projection stage.
parallel applications independent of the computing platforms (a cluster of workstations, supercomputers, or even the computational grid). In the work reported here, we have implemented the parallel iterative reconstruction method using standard C and MPI. We have tested the functionality of the application in supercomputers (the IBM SP2 at the San Diego Supercomputer Center) and the cluster of workstations at the NCMIR (National Center for Microscopy and Imaging Research). The performance results described and analyzed here were measured in the cluster. Such a cluster consists of 30 computers (uni- and dual-Pentium III) which involve 50 processors and a total of 17 GB of RAM distributed across them. The computers in the cluster are switched by fast ethernet and myrinet networks.

4. Results

The experiments that have been carried out had two aims. First, the effective speed-up and the computation times that our parallel approach yields were computed so as to evaluate its efficiency. Second, we intended to do a fair comparison between WBP and the component averaging methods in terms of the quality in the reconstructions. Here we will show the evolution of the iterative methods as a function of the iteration. Finally, the results from the application to experimental mitochondrial data will be shown.

We have tested WBP, CAV, and BICAV under different parameters:

- Block sizes for component averaging methods: 1, 10, 30, 50, and 70. We have considered that the size of a block should be multiple of the size of the projection images so that all the pixels of the same projection image belong to the same block.

- Blob parameters. We have tested five different blobs with radii 1.25, 1.5, 2.0, 2.25, and 2.795. The parameter $m$ was set up to 2 for all of them. The values for the parameter $x$ used for the different blob radii have been 3.6, 6.4, 10.4, 3.5, and 4.7, respectively. The values for the parameter $x$ were analytically computed according to the anti-aliasing condition that the frequency spectrum of the blob must be zero at the sampling frequency (Le-witt, 1992; Matej et al., 1996b).

Those parameters provided five blobs with different FWHM: 1.17, 1.20, 1.33, 2.15, and 2.45, respectively, relative to the sampling interval.

- Relaxation factors for the iterative methods.

The concept of resolution that has been used in this work refers to the maximum resolution attainable in the reconstructions, according to the spectral features of the blob used. This resolution is computed directly from the FWHM of the blob. In this work, the results from WBP have been filtered using the Fourier transform of the blob as the low-pass filter in order to do a correct comparison with the corresponding iterative methods.

4.1. Evaluation of the parallel approach

The efficiency of any parallelization approach is usually evaluated in terms of the speed-up (Wilkinson and Allen, 1999). The speed-up is defined as the ratio between the computation time required for the application to be executed in a single-processor environment and the corresponding time in the parallel system. The effective speed-up that we have obtained with our parallel approach for iterative methods is nearly linear with the number of processors in the parallel system.

On the other hand, in order to compare component averaging methods with WBP in terms of the computational burden, we have measured the time required for every iteration. We have taken such measures for different reconstruction sizes ($256 \times 256 \times 256$, $512 \times 512 \times 512$, $1024 \times 1024 \times 200$), different number of tilt angles (61 or 70), and different blob sizes, obtaining similar relative behaviors. Fig. 5 shows the results obtained for a $512 \times 512 \times 512$ reconstruction from 70 projections, using a blob with radius 2.0 (bars in light gray) and 2.795 (bars in dark gray). In Fig. 5, it is clearly observed that WBP is the lowest resource-consuming method, since it requires 235 s for the whole reconstruction. However, one iteration of the component averaging methods requires nots much more time (324–358 and 438–496 s for blobs with radii of 2.0 and 2.795, respectively). Taking into account that these iterative methods are really efficient and yield suitable solutions after a few iterations (as will be shown later), these computation times make component averaging methods real alternatives to WBP, according to the trade-off of computation time vs quality of the reconstruction.

It should be noted that, in iterative methods, the larger the number of blocks into which the projections are split, the more communications are needed (there are two communication points per block) and, as a consequence, the time required for one iteration should be larger. However, the latency hiding approach that we have used in this work succeeds in hiding or minimizing this increasing time. In Fig. 5 it can be observed that as the number of blocks increases, the time per iteration is slightly larger (from 324/438 s in CAV to 358/496 s in BICAV with 70 blocks, using a blob with radius 2.0/2.795, respectively). But this slight increase in time is due mainly to the communication start-up time, which is not avoidable for any latency hiding approach.

4.2. Evaluation of component averaging methods

The comparison between WBP and the component averaging methods has been carried out in terms of the quality in reconstructing artificial volumes (phantoms).
In this work we have designed a phantom resembling the mitochondria. It consists of hollow cylinders representing the membranes and a set of solid cylinders simulating the cristae. The cristae are embedded in a region of intermediate density resembling the mitochondrial inner matter. We have also tested different sizes of the phantom, with projections simulating a single-tilt axis geometry with 61 and 70 tilt angles, and under different noise conditions: SNR 5 and 2.

The quality of the reconstructions has been evaluated based on the structural consistency figures of merit (FOMs) described in Sorzano et al. (2001) and the relative error (Censor et al., 2001b). Table 1 presents the expressions corresponding to all the FOMs that have been used. In this work, the FOMs were measured over a region of interest that fit the set of the cylinders resembling the cristae. We computed the FOMs for reconstructions with component averaging methods as a function of the iteration. The FOMs were also computed for the WBP results. Due to the noise conditions, it was found to be convenient to apply a low-pass filter to the WBP results. The transfer functions of the different blobs were used as such a low-pass filter so as to do a fair comparison with the blob-based iterative methods.

We have obtained similar curves for all the FOMs and phantom sizes. Here we show only the results obtained for the FOM scL2 (i.e., the square Euclidean distance between the reconstruction and the phantom) for the $256 \times 256 \times 256$ phantom and using 61 projections. As mentioned at the beginning of this section, we have tested the algorithms for different values of block, blob radii, and relaxation factor. Fig. 6 presents the evolution with the iterations of the FOM values corresponding to WBP, CAV and BICAV for SNR = 2 for different parameters. In all of these figures, the color of the curve represents the blob involved in the reconstruction: red, green, blue, and magenta color represent blobs with radius, respectively, 1.25, 1.5, 2.0, and 2.25. In the case of WBP, the FOM value is represented by a constant curve, and the color denotes the blob whose transfer function was used as the low-pass filter.

Fig. 6a shows the results from WBP, CAV using a relaxation factor with a value of 2.0 and, finally, BICAV using 61 blocks (i.e., 1 tilt angle per block) and a relaxation factor of 0.25. The behavior of WBP strongly depends on the postreconstruction filtering due to the high noise level. It is apparent that the strongest low-

Table 1

<table>
<thead>
<tr>
<th>Structural consistency FOMs used to evaluate the reconstruction algorithms</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\text{scL2}(R) = 1.0 - \frac{1}{N_B} \sum_{\rho,k} \left( \frac{\rho_i - r_j}{2} \right)^2$</td>
</tr>
<tr>
<td>$\text{scL1}(R) = 1.0 - \frac{1}{N_B} \sum_{\rho,k} \frac{</td>
</tr>
<tr>
<td>$\text{scL1}(R) = 1.0 - \frac{1}{2}</td>
</tr>
<tr>
<td>$\text{scL1}(R) = 1.0 - \frac{1}{2}</td>
</tr>
<tr>
<td>$\text{scL1}(R) = \frac{\sum_{i \neq k}</td>
</tr>
</tbody>
</table>

Note. $N_B$ is the number of samples in the region $R$. The terms $\rho_i$ and $r_j$ represent the density value of the $i$th sample of the phantom and the reconstruction, respectively. The average density of the phantom and the reconstruction over a region $R$ are given by $\mu_{p,k}$ and $\mu_{r,k}$, respectively. The terms $\sigma_{p,k}$ and $\sigma_{r,k}$ represent the standard deviation of the density in the phantom and in the reconstruction, respectively, over the region $R$. The FOMs are defined so that they yield a measure of quality for a specific region $R$ of the volume. $\text{scL2}(R)$ represents the square Euclidean distance (L2 norm) between the phantom and the reconstruction, measured over the region $R$. $\text{scL1}(R)$ represents the L1 norm. $\text{scL1}(R)$ and $\text{scL1}(R)$ denote the difference in average and standard deviation, respectively, between the phantom and the reconstruction. Finally, $\text{scC}(R)$ represents the relative error over the region $R$.  

---

Fig. 5. Evaluation of the parallel approach. The computation times required by WBP and component averaging methods for a $512 \times 512 \times 512$ reconstruction from 70 projection images using different blob radii are shown. Component averaging methods were tested for different number of blocks.
pass filter (the blob with a radius of 2.25, represented in magenta) yields the best result according to the \( \text{scL2 FOM} \). CAV is a stable algorithm, that uniformly progresses upward, independent of the blob. CAV outperforms WBP after 6–8 iterations with either blob. Finally, the version of BICAV shown here uses the maximum number of blocks as stated above. Here, the relaxation factor has been set up to a small value in order to stabilize the algorithm. Under these conditions, BICAV becomes superior to WBP after 2–3 iterations. However, BICAV may become unstable after 10 iterations, yielding decreasing FOM values. The instability strongly depends on the blob chosen: the narrower the blob, the more unstable the algorithm, as shown in Fig. 6a.

The relaxation factor is of great importance, as shown in Fig. 6b. If the relaxation factor is too high (in this case 1.0), the BICAV algorithm rapidly achieves the top FOM value (in three or four iterations), but soon becomes unstable. The slope of the FOM then turns around, yielding a decreasing curve after the first four or five iterations. Again, the selection of the blob is important to control the decay of the curve. Wider blobs yield more stable solutions.

The tendency of the algorithm may be controlled by tuning the different parameters: relaxation factor, number of blocks, and blobs. Fig. 6c shows the steady curve provided by BICAV when 10 blocks and a relaxation factor of 0.4 are used. As an example, we have obtained results (not shown here) in which BICAV with (i) 61 blocks and a relaxation factor of 0.25, (ii) 30 blocks and a relaxation of factor of 0.4, and (iii) 10 blocks and a relaxation of 1.0 yielded very similar behaviors.

When the noise conditions are more relaxed (SNR = 5, in this work), the algorithms have a relative behavior that is similar to those just described (graphs...
not shown here). However, it should be highlighted that WBP yields much better results (FOM values of around 0.983 for any low-pass filter) than with extremely low SNR. But still, CAV and BICAV clearly outperform WBP after a number of iterations in the range $[1, 10]$, depending on the parameters.

Fig. 7 shows visual results of one of the slices along the tilt axis of the reconstruction of a $340 \times 340 \times 340$ mitochondria phantom from 70 projections. A view of the phantom and the slice to be reconstructed are at the left of the figure. In the upper row, the results after four iterations of BICAV (70 blocks and a relaxation factor of 1.0) are shown for three blobs with different radii (from left to right: 1.25, 2.0, and 2.795). In the lower row, the corresponding results from WBP and postreconstruction filtering are shown. The resolution of a BICAV result and that of the WBP result just below it are equivalent. It may be clearly observed that BICAV yields more regularized results than any WBP result for all of the blobs. WBP requires the low-pass filtering provided by the widest blob (radius $= 2.795$ and FWHM $= 2.45$) to yield a result similar to those provided by BICAV. However, the BICAV results with narrower blobs (radii 1.25 and 2.0, with FWHM 1.17 and 1.33, respectively) have better resolution.

4.3. Application to electron tomography of mitochondria

Finally, we have applied the BICAV and WBP methods to experimental mitochondria data obtained from HVEM and prepared using photooxidation procedures. Seventy projection images (with tilt angles in the range $[-70, +68]$) were combined to obtain the reconstructions. The projection images were $1024 \times 480$, and the volume was $1024 \times 480 \times 256$. We have tested the algorithms under different conditions, but here only the most significant results are presented. In general, the results that have been obtained with those data follow the pattern exhibited by the phantoms under extremely noisy conditions (SNR = 2).

A montage showing one z-section of the volume reconstructed with the different methods is presented in Fig. 8. Fig. 8a shows the results coming from WBP (with a postreconstruction filtering equivalent to a narrow blob with radius 1.25). Fig. 8b shows the reconstruction obtained from BICAV with 70 blocks after 4 iterations with a relaxation factor of 1.0 and a blob with radius 1.25 (those parameters were chosen in view of the fast convergence rate exhibited in our work with phantoms, see Fig. 6b). Since these two results are based on the same blob, they are directly comparable in terms of the

---

Fig. 7. Visual comparison between WBP and component averaging methods. At the left, a sketch of the phantom and the slice under consideration. In the upper row, the results from four iterations of BICAV (70 blocks and a relaxation factor of 1.0) are presented for three blobs with different radius (from left to right: 1.25, 2.0, 2.795). In the lower row, the corresponding results from WBP and postreconstruction filtering are shown.
maximum resolution attainable. We have also made tests with CAV, but the solutions still are blurred after 30–50 iterations, due to its relatively slow convergence rate compared to BICAV.

Fig. 8 clearly shows that blob-based BICAV yields a solution much cleaner than WBP and, moreover, at the same resolution level (FWHM = 1.17). The excellent behavior shown by BICAV under such noisy situations comes from the regularization provided by the blobs, even though the blob radius that has been used for results in Fig. 8 is very small. However, WBP provides a “noisy” solution due to the high noise level in the experimental projection images, even after the postreconstruction filtering.

We have also tested wider blobs, and the results prove to be smoother than those presented. Also, we have tested different relaxation factors for BICAV. We have noted that when relatively high relaxation factors are used for many iterations, the reconstructions turn out to be darker, have less gray levels, and consequently finer

Fig. 8. Comparison of WBP and component averaging methods in an experimental application. Seventy experimental HVEM images from mitochondria prepared using photooxidation procedures were used to compute the reconstructions. One of the slices along the Z axis is shown. (a) Result from WBP and a postfiltering equivalent to a blob with radius 1.25. (b) Result from BICAV with 70 blocks after four iterations with a relaxation factor of 1.0 and a blob with radius 1.25.
details vanish. In contrast, if the relaxation factor is conservatively selected, the algorithm requires more iterations to achieve a good solution.

Regarding the computation times, the reconstructions were done in the cluster of workstations at the NCMIR, using 50 processors. For the results shown in Fig. 8, WBP took around 200 s of computation time to obtain the solution, whereas BICAV took around 1300 s (which involves around 325 s per iteration).

5. Discussion and conclusions

In this work we have analyzed the application of blob-based series expansion methods in electron tomography of complex biological specimens, with a special emphasis on the computational perspective. First, we have made use of efficient iterative methods to tackle the problem of image reconstruction. On the other hand, HPC techniques have been applied so as to face the high computational demands and take advantage of parallel systems. The results that we have obtained clearly show that the combination of those new iterative methods and HPC is well suited to tackle the reconstruction of large biological specimens in electron tomography, yielding solutions in reasonable computation times.

A parallel approach for the iterative algorithms has been devised that has a speed-up nearly linear with the number of processors in the system. This approach have also proved to be very efficient in dealing with the communications among the processors. The results (see Fig. 5) indicate that the latency due to communications is almost completely hidden by overlapping computation and communication. This parallel strategy allows the iterative methods to take between 5 and 8 computation minutes per iteration in the reconstruction of a $512 \times 512 \times 512$ volume. The parallel approach is also exploited for WBP (with the advantage that communications are no longer needed) in such a way that a WBP result of the same size is obtained after nearly 4 computation minutes. In this way, HPC is providing cost-effective solutions to grand challenge problems (for instance, reconstructions of $2048 \times 2048 \times 2048$) which are currently unapproachable by uni-processor systems due to the computational requirements.

The new iterative reconstruction methods that we have applied here are very efficient, providing least-squares solutions in a very few iterations. Specifically, BICAV with a large number of blocks yields good reconstructions in a number of iterations in the range $[1, 10]$, depending on the parameters. BICAV could produce a solution after a computation time of 5–8 min for a $512 \times 512 \times 512$ volume for one iteration, compared to 4 min needed by WBP.

On the other hand, the use of blobs has a twofold benefit. First, blobs are more optimal than voxels from the computational point of view, since they allow the use of look-up tables to speed up the forward- and backward-projection stages. Second, blobs provide the reconstruction algorithms with an implicit regularization mechanism which makes them well suited to noisy environments. As a consequence, the solutions yielded by blob-based iterative methods are smoother than those yielded by WBP, but with relatively unimpaired resolution. In particular, under extremely noisy conditions, these types of algorithms clearly outperform WBP. In those situations, WBP requires a strong low-pass filtering postreconstruction stage which limits the maximum resolution attainable (see Fig. 7).

Regularized iterative reconstruction methods have additional and interesting implications in data postprocessing, particularly segmentation. Segmentation is the process that allows one to identify and dissect components of the reconstructed volume. It is particularly useful in facilitating interpretation and measurement of the features of large complex structures obtained from electron tomography. Generally, the segmentation of the complex structural components is accomplished by means of manual tracing (Perkins et al., 1997). However, there exist automatic segmentation techniques whose application in this field is hampered by the noise present in the reconstructed volume. In this sense, regularized reconstruction methods are more suitable for these segmentation techniques, since they reduce noise in the reconstructions. This proves to be a major advantage of regularized reconstruction methods, such as those presented in this work, for future segmentation efforts.

The performance of the algorithms in the quality of the reconstructions has been objectively measured via some structural consistency FOMs already described (Censor et al., 2001b; Sorzano et al., 2001). We have computed the FOMs as a function of the iteration in the reconstructed volume. In this sense, regularized reconstruction methods are more suitable than those presented in this work, for future segmentation efforts.

The set of parameters associated with the method should be properly tuned according to the specimens under consideration by following, for instance, the methodology described in Marabini et al. (1997). Nevertheless, we have found that it is still possible to control the algorithms by a set of rules of thumb, derived from our own experience:

- The convergence rate in the iterative algorithms has been found to strongly depend on the relaxation factor and the number of blocks. Both of them influence
the algorithm in the same way: The higher value, the faster initial convergence rate, but the sooner the tendency turns around. An extreme situation would involve the use of a relaxation factor of 1.0 and as many blocks as tilt angles. In this situation, using a very few number of iterations (1–4) is sensible (see Fig. 6b). More conservative options (see Figs. 6a and b) would involve the use of a relaxation factor in [0.25, 0.4] and a number of blocks around half the tilt angles (25–40).

• The number of iterations in the algorithms should be selected depending on the relaxation factor and number of blocks. CAV requires at least 30 or 40 iterations. In general, BICAV needs between 1 and 20 iterations. The higher the relaxation factor and the number of blocks, the fewer the iterations.

• The parameters of the blob control the regularization mechanism in the algorithm. The wider the blob, the more robust to noise, but the less detail. Narrower blobs yield more structural details, but at the risk of artifacts in the presence of high levels of noise.

In addition, blob parameters have been found to control the stability of the BICAV algorithm. When high relaxation factors and many blocks are used, wider blobs help stabilize the FOM curves, whereas narrower blobs produce FOM curves with rapid decays (see Fig. 6). On the other hand, in the initial iterations of the algorithm (see Figs. 6a and b) and in those situations where the other parameters have been conservatively chosen (see Fig. 6c), the influence of the blob in the FOM is nearly negligible.

Finally, the application of the algorithms to experimental HVEM images of mitochondria using photooxidation techniques shows that the results obtained with phantoms approach the real situation. In that sense, the usefulness of the guidelines on the use of the blob-based component averaging methods on real data has been verified.

Nevertheless, in practice applications, there are still open questions that should be further investigated in future works, such as image normalization, image alignment, and others. Some of those problems could be addressed from an iterative perspective and, in that sense, the fact that regularized reconstructions are well suited to postprocessing stages may be central.

Acknowledgments

The authors thank Dr. G. Perkins who kindly provided the real mitochondria data and contributed valuable comments on the reconstructions. The authors also thank C.O.S. Sorzano and J.G. Donaire for fruitful discussions during the work. The work has been partially supported through grants from the Spanish CI-

References


