

Parallelization of gradient-based iterative image reconstruction scheme

S. Bartel, G. Abdoulaev, and A.H. Hielscher

State University of New York - Downstate Medical Center, Department of Pathology,
450 Clarkson Avenue, Brooklyn, NY 11203, USA
ahielscher@netmail.hscbklyn.edu

Abstract: To increase the speed of gradient based iterative image reconstruction schemes we have implemented parallel execution algorithms on a heterogeneous cluster of workstations. A dynamic and a static process allocation scheme are introduced and discussed.

©1999 Optical Society of America

OCIS codes: (170.6960) Tomography; (170.3010) Image reconstruction techniques

(Document ID OSA_Miami'00: 14573)

1.0 Introduction

Image reconstruction schemes for optical tomography are computationally very demanding. Since no generally applicable analytical inversion algorithm exists, researchers have focused on iterative reconstruction methods. Within this class, so-called gradient-based reconstruction (GIIR) schemes have attracted increasing interest [1-3]. These schemes approach the reconstruction problem by minimizing an objective function that reflects the quality of a reconstructed image. The minimization process typically requires many so-called forward calculations, in which a light propagation model is used to simulate a set of detector readings for a given distribution of optical parameters. The most limiting factor concerning speed appears to be the time it takes to solve the forward problem. In this work we first analyze in detail the computational expenses of various parts of GIIR algorithms. Based on this analysis we develop parallel algorithms for a heterogeneous cluster of workstations connected through a TCP/IP network and thus considerably increase the speed of the reconstruction process.

2.0. Methods

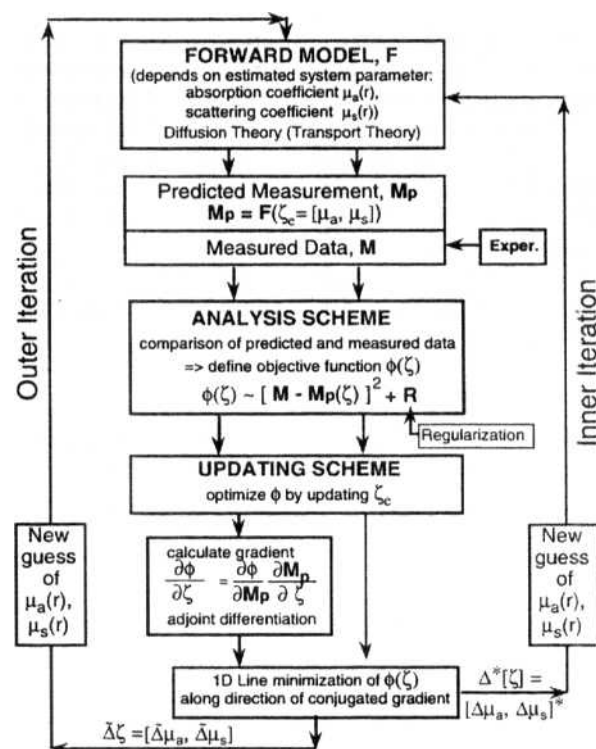
2.1. Gradient based iterative image reconstruction

To understand how GIIR schemes can be parallelized we first review the basic structure of these schemes. Recently, so-called gradient-based iterative image reconstruction (GIIR) schemes have found increasing attention. As in other reconstruction schemes the goal of GIIR algorithms is to reconstruct the optical properties, e.g., absorption, $\mu_a(\mathbf{r})$, and diffusion, $D(\mathbf{r})$, coefficients, at positions \mathbf{r} inside a medium, from a given set \mathbf{M} of measurements on the circumference of the medium. Our scheme has three major components:

(1) **Forward Model:** This model is a theory or algorithm that predicts a set of measured signals, \mathbf{U} , based on the position of the light source and the spatial distribution of optical properties $\zeta = (c\mu_a(\mathbf{r}), D(\mathbf{r}))$. In this work we employ a time-dependent diffusion model.

(2) **Analysis Scheme:** Here an objective function, ϕ , is defined, which describes the difference between the measured, \mathbf{M} , and predicted data, \mathbf{P} . A simple example is the least-square error norm, also called χ^2 -norm,

$$\phi(\zeta) = \sum_s \sum_d (M_{s,d} - P_{s,d}(\zeta))^2 / \sigma_{s,d}^2, \quad (1)$$



where the sums are taken over all source and detector positions s and d , respectively.

(3) **Updating Scheme:** Once the objective function is defined, the task becomes to minimize ϕ . This is accomplished in two sub-steps. First the gradient of the objective function $d\phi(\zeta)/d\zeta$ is calculated by means of reverse-adjoint differentiation. Secondly, given the gradient an iterative 1-dimensional line-minimization in the direction of the gradient is performed. This step is referred to as *inner iteration* and consists of several forward calculations in which the parameters ζ are varied and the objective function $\phi(\zeta)$ is evaluated. Once the minimum along the line is found, a new gradient is calculated at this minimum (*outer iteration*) and another line-minimization is performed, now along a different direction in the ζ -space. These steps are repeated until a distribution ζ is found for which $\phi(\zeta)$ is smallest.

2.2. Parallelization

The appropriate decomposition of the original problem into smaller subtasks is most crucial to every parallel implementation of a given algorithm. We identified the forward model, a finite-differences algorithm, solving the diffusion equation, as the most time-consuming portion of our GIRR reconstruction scheme. Approximately 95 % of the computational time are spent here, with each forward calculation requiring in the order of 0.1 to 10 seconds (depending on the size of the finite-difference grid) on a 450 MHz Intel PII processor running Linux 6.1.

Experimental data typically consists of several sets of detector readings corresponding to different source positions. To match these reading with the simulation and evaluate the objective function (1), each source present in the experiment, requires a separate forward-calculation. Hence, the problem of determining (1) for N sources quite naturally breaks up into N mutually independent forward calculations. We can rewrite Eq. 1 as

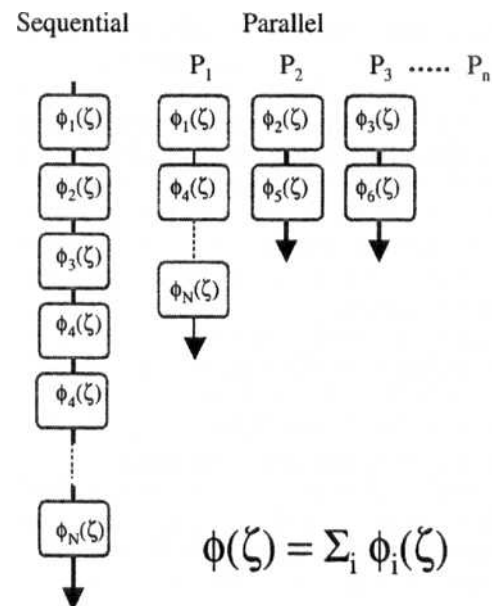
$$\phi(\zeta) = \sum_s \sum_d (M_{s,d} - P_{s,d}(\zeta))^2 / \sigma_{s,d} = \sum_{s=1}^N \phi_s \quad (2)$$

The objective function ϕ is obtained by summing over all source-detector combinations. Instead of handling all N sources sequentially, we may therefore share the total load amongst up to N processors, by assigning each processor a distinct set of sources to evaluate and then adding the partial results. This is where the algorithm is suitably parallelized.

We implemented two schemes to perform load sharing across a network of heterogeneous workstations. First we employed a *dynamic* algorithm that takes into account different processor speeds, and second, a *static* algorithm that assigns a preset number of sources to each processor.

The *dynamic scheme* employs a single dedicated processor to schedule and assign different sources to individual processors. This main processor responds to requests from others by assigning one out of N available sources to the calling processor, while keeping track of the tasks already assigned.

All remaining processors, upon entering the forward calculation, inquire the main processor about a task due, namely a single source to evaluate, and perform the associated forward calculation. As soon as a calculation has been completed, another request is posted and the above step repeated, until no more sources are available. At this point, all processors are synchronized and the partial results summed across the cluster.



Decomposition of objective function ϕ into single-source objectives ϕ_i , corresponding to source i turned on. The gradient $d\phi/d\zeta$ is decomposed similarly.

The static scheme is a somewhat simpler approach, in which all processors are treated equally, by distributing the total number of sources across participating processors $P_1, P_2, P_3, \dots, P_N$ in a predefined manner regardless of individual speeds. Assume, n processors are involved in the calculation, then we assign sources $1, n+1, 2n+1, \dots$ to processor 1, sources $2, n+2, 2n+2$ to processor two and so forth (Fig. right). In general, processor i out of n performs forward calculations for the sources $i, n+i, 2n+i, \dots$. When all processors have completed their share of forward calculations, the overall result is again determined by summing over all partial results.

The actual communication between processors, i.e. distributing data, organizing sub-tasks and gathering results, is handled by the so-called Message Passing Interface (MPI). This standard for inter-processor communication in distributed memory architectures, introduced in 1992/93, provides a system-independent interface for sending and receiving messages across different hardware platforms. It is freely available for a variety of systems.

3. Results and Discussion

We tested both, the dynamic and static scheme on a cluster of workstations, consisting of three intel-based dual-processor machines (350 Mhz & 450 MHz) running Linux and 2 SGI's (Crimson, 2x R4400 150MHz, Origin, 2x R10000 180 MHz) running IRIX 5.3 / 6.1. The machines are connected by a 10 MB Ethernet, twisted pair network. At this point, only passive routers were used, future work will make use of active switches and a local 100 MB subnet to improve bandwidth and decrease latency within the cluster.

We found that the main advantage of the dynamic load sharing is, that faster processors are enabled to post and process several requests while others might still be evaluating their first source. This balances the total work, according to the various processor speeds. However, using standard Ethernet connections we observed a significant amount of idle time while processors wait for a task to be assigned by the main processor. This is mainly due to long latencies of the TCP/IP, which may exceed the millisecond range in busy networks.

The advantages of the dynamic load sharing only become apparent at large problem setups, when the average time spent in a single forward calculation exceeds the latency of the network. Moreover, since one processor is dedicated to managing the task-assignments, an increase in performance does not occur unless more than two processors are employed.

The less sophisticated, static assignment of tasks to individual processors avoids all unnecessary communicational overhead and performs particularly well in a homogeneous cluster of workstations. With the number of processors being a multiple of the number of sources N , we obtained a degree of parallelization beyond 90%, i.e. using 4 processors resulted in an performance increase > 3.6 . These results are almost independent of the dimension of the reconstruction problem, in contrast to the dynamic scheme above.

4. Summary

We have developed and implemented two types of parallel execution schemes for iterative gradient-based image reconstruction on a heterogeneous network of workstations.

The static load sharing makes almost optimal use of the available resources, if used in a homogeneous cluster and the number of sources being a multiple of the number of processors. In heterogeneous networks, the dynamic scheme performs better, since it takes different processor speeds into account and avoids unnecessary waiting for slower participants to complete their share. However, the associated communication overhead can be justified only in reconstruction problems that are sufficiently large so that the time necessary to complete a task is much larger than the latency for communicating the result. Future demands on image reconstruction will certainly meet this condition and make the dynamic allocation scheme the preferable alternative.

5. References

- [1] S. R. Arridge and M. Schweiger, "A gradient-based optimisation scheme for optical tomography," *Optics Express* 2 (6), 213-226 (1998).
- [2] A.H. Hielscher, A.D. Klose, K.M. Hanson, "Gradient-based iterative image reconstruction scheme for time-resolved optical tomography," *IEEE Tran.s. Med. Imag.* 18, 262-271 (1999).
- [3] R. Roy and E.M. Sevick-Muraca, "Truncated Newton's optimization scheme for absorption and fluorescence optical tomography: Part I Theory and formulation, *Optics Express* 10, 353-371 (1999).