

Acceleration of Maximum Likelihood Estimation for Tomosynthesis Mammography

Juemin Zhang, Waleed Meleis, and David Kaeli
Northeastern University
Electrical and Computer Engineering
Boston, MA 02115
{jzhang, meleis, kaeli}@ece.neu.edu

Tao Wu
Radiology Dept.
Massachusetts General Hospital
Boston, MA 02114
twu2@partners.org

Abstract

Maximum likelihood (ML) estimation is used during tomosynthesis mammography reconstruction. A single reconstruction involves the processing of high-resolution projection images, which is both compute-intensive and time-consuming. This workload is presently a bottleneck in the accurate diagnosis of breast cancer during screening. This paper presents our parallelization work on an ML algorithm using three different partitioning models: no inter-communication, overlap with inter-communication and non-overlap model. These models are evaluated to obtain the best reconstruction performance given a range of computing environments with different computational power and network speed.

Our test results show that the non-overlap method outperforms the other two methods on all five computing platforms evaluated. This parallelization of ML has enabled tomosynthesis to become a viable technology in the breast screening clinic, reducing reconstruction time from 3 hours on a PentiumIV workstation to 6 minutes on a 32-node PentiumIV cluster.

1 Introduction

Presently, *mammography* is the most effective technique used in the detection of breast cancer. A digital mammogram is an image projection produced by a distribution of x-ray attenuation through breast tissue. Some information in three-dimensions may be lost when it is projected onto a two-dimensional plane. The structural information of a breast can become blurred due to overlapped and superimposed tissues. Overlapped tis-

sues inside a breast can obscure a cancer, which causes over 30% of the cancers to be missed when using traditional mammographic techniques [5]. Superimposed normal tissues can sometimes look like an ill-defined shaped tumor in a mammographic image. This can cause a large number of callbacks due to the limitations of current mammography techniques.

Tomosynthesis provides structural information associated with a 3D object in layered images [3]. The internal structural information of the object is computed from a set of discrete x-ray projections obtained at different angles.

Currently, tomosynthesis mammography is under investigation at Massachusetts General Hospital (MGH). The goal of this technique is to address the breast tissue superimposition problem [10, 12, 11, 13]. A total 11 x-ray digital mammograms are acquired by moving the x-ray source over a range of 50° , while the object and digital x-ray detector are held stationary. To reduce the amount of x-ray radiation exposure to the patients, tomosynthesis uses a lower dose of x-ray than conventional mammography uses during x-ray image acquisition. After 11 x-ray image projections are acquired, tomosynthesis relies on an image reconstruction algorithm to recreate the 3D structure of a breast and to enhance the visibility of features which aid doctors and physicians in the detection and diagnosis of tumorous tissue.

An iterative maximum likelihood (ML) algorithm was developed in the clinical study of tomosynthesis mammography [13]. The reconstructed images have proven to be very effective in distinguishing the overlapped tissues of a breast in the clinical study performed at MGH. Tumorous tissues that were missed using traditional mammography due to overlapped tissues can be clearly identified in the layered tomosynthesis images. Even since the introduction of tomosynthesis in 2005,

the number of false-positive callbacks has been reduced.

High quality and high-resolution image reconstruction are needed in practice. A high-definition x-ray detector panel used by the prototype tomosynthesis system at MGH yields 1900×2304 pixels. After reconstruction, the image can reach the size of 1600 pixels in width, 2304 pixels in length, and 50 layers in depth (where each layer is 1mm thick). A single 3D image reconstruction will consume over 500MB of disk space, over 2GB of memory space (during reconstruction) and more than 3 hours of execution time as run on a Pentium-IV.

The number of clinical cases that require tomosynthesis reconstructions is overwhelming even for several fast workstations. The efficiency and usability of tomosynthesis mammography are limited due to the complexity of the image reconstruction.

1.1 Reconstruction method

The ML algorithm defines the probability likelihood function as:

$$L = P(Y|u)$$

, where Y represents the set of acquired x-ray projections and u stands for attenuation coefficients of a 3D volume. The maximized solution of $L = P(Y|u)$ can produce an estimate of the 3D volume that projects an image most similar to Y . However, most analytical solutions to the likelihood function are intractable. Iterative maximum likelihood estimation is a commonly used algorithm to solve such problems.

The basic idea of employing an iterative reconstruction is to continually correct the initial guess of the 3D volume of an object until the reconstruction closely resembles the original object. During each iteration, the reconstruction proceeds in two phases: 1) a forward projection, followed by 2) a backward projection (see Figure 1).

Given an initial estimated 3D volume of an object, the forward projection simulates how x-ray beams are absorbed when they are traveling through the object, and creates estimated projection images. During the backward projection phase, each value in the 3D volume is corrected based on differences between the estimated projections and the actual x-ray projections.

Values within the 3D volume, representing features of the object, are strengthened at the corresponding geometry locations during each iteration. This process typically takes 8 to 10 iterations to optimize the image quality, so that most of structural information can be assessed qualitatively by doctors and physicians.

The complexity of ML estimation increases linearly with the size of the detector panel, the number of reconstruction iterations, and the depth of the 3D volume. The reconstruction assigns each pixel an unsigned-short type and adapts integer computing to enhance its performance as well as to decrease memory demands. However, the x-ray tracing technique used in the forward and backward projection process requires a lot of floating-point computing to calculate the x-ray beam's trace.

2 Parallelization

One of the goals for this research is to accelerate the image reconstruction process and make tomosynthesis more practical to use in the clinic. Analysis of different approaches can also provide options in building the tomosynthesis computing components, as well as a portable high-performance solution. The parallelization approach for accelerating ML reconstruction needs to be general enough to adapt easily to different parallel and distributed systems. Although there is no real-time deadline imposed on performance, the reconstruction time of an average sized breast volume should be less than 15 minutes, so that the whole screening and diagnosis process can be completed during the same patient visit.

2.1 Segmentation method

Image reconstruction of the ML algorithm was originally implemented at MGH [13]. In an earlier profiling study of the sequential ML implementation, we discovered that 40-60% of the execution time is spent on cache misses and page faults. Data partitioning the large dataset can help to reduce the number of cache misses and page faults and improve the performance by parallel execution. Our parallelization implements data partitioning by using message passing.

Projected images and 3D volumes of an object are partitioned into multiple segments, which are processed in parallel on a parallel clustered system. Selecting the proper sized partition should help to expose parallelism, and should limit the amount of data required to be exchanged between processing nodes. The fewer data dependencies found across different partitions, the less communication.

During the reconstruction process, the 3D volume data and the projection images will reference each other when accessing boundaries of the x-ray beam. The forward projection simulates how x-rays are absorbed, and estimates in the projected images where the x-ray beams

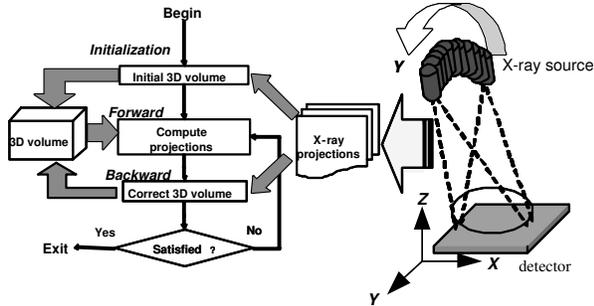


Figure 1. Tomosynthesis reconstruction and projection images acquisition process.

pass through. The backward projection corrects data in the 3D volume, and then this correction is computed in all projections. Therefore, if partition edges could be aligned with the x-ray beam's direction, there would be fewer internode data references, as seen in Figure 2.

As the x-ray source shifts, there is more than one x-ray beam traveling through a point inside the 3D volume of an object. Even when we partition a 3D volume along the direction of the x-ray beam, we can not avoid cutting through another x-ray beam because the beam moves. Since the x-ray source moves in the Y-axial direction over 50° when taking x-ray projection images, (see Figure 1), partitioning along the Y-axial direction will lead to the least cross-segment data references. Thus, we choose to project each image and 3D volume in the Y-axial direction.

In Figure 2, the 3D volume data is partitioned in a cone shape, as each layer of the 3D volume is partitioned evenly on the Y-axial direction.

As the x-ray source shifts, some x-ray beams will travel through the edges of two neighboring segments. Data whose geometry locations lie on or next to such x-ray beams will be referenced by both segments. When a new value is assigned to the data in one segment after computing the forward or backward projection, the other segment also needs to be updated.

Two subregions are added to each segment of the 3D volume, with only a single extension being added for the two segments located on the boundaries of the object. The extension region of a segment contains all data which will be referenced in the reconstructed segment. The estimated projections and other temporary data used in reconstruction have corresponding extensions too. The size of the extensions on each segment can be pre-calculated before reconstruction, because the

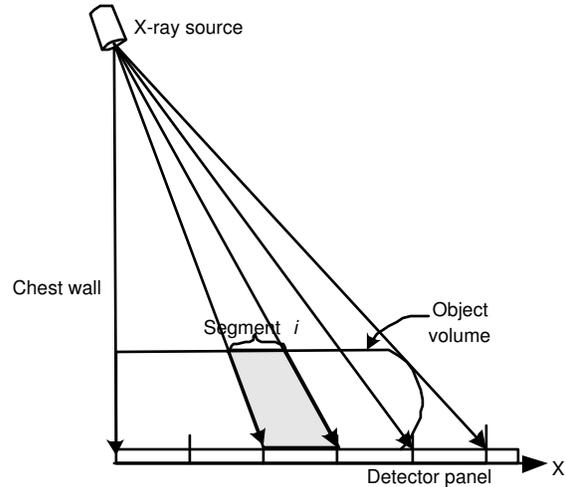


Figure 2. 3D volume segmentation method.

size only depends on the geometry of the segment.

After the forward projection has been computed, data in the boundary region of the estimated projection needs to be updated. After the backward projection has been computed, data in the extension region of the 3D segment needs to be updated too. To update extension regions of a segment, we introduce three models:

- No inter-communication model, which computes both extension regions locally.
- Overlap with inter-communication model, which computes part of the extension locally and transfers the rest from neighboring segments.
- Non-overlap model, which transfers data in both extension regions from neighboring segments.

The three methods differ from each other by how extension regions are updated: computation, inter-process communication, or both.

2.2 Non inter-communication model

The no inter-communication model computes two extension regions for each 3D segment (see Figure 3), instead of acquiring them from neighboring segments which also compute the values in the extension regions. The corresponding extension region projections are also computed locally. This model treats the segment and its extension regions as one consolidated object, with the exception that its geometry is based on the partition of the original object.

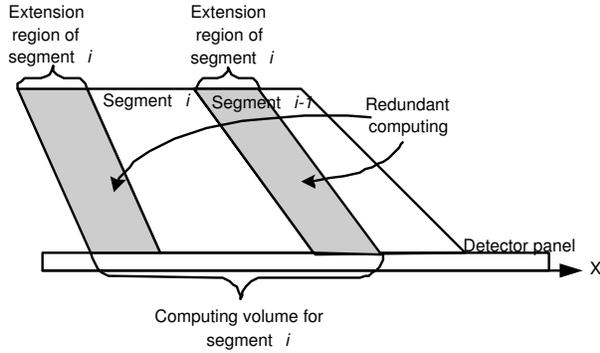


Figure 3. Non inter-communication model

After all reconstruction iterations are complete, all reconstructed segments are reassembled together, and data in extension regions is ignored. Considering that the overhead due to inter-process communication could be very expensive on some slow networks, this model can eliminate the inter-process communication by allowing each partitioned segment to be reconstructed independently. This method is a coarse-grained parallelization which can be implemented as a task-level parallel application and deployed on most of distributed systems.

During each reconstruction iteration, the reconstruction of each 3D segment and its extensions are independent from computations for other segments. We can assume that there are no contents outside the extension region. Values calculated by the ML algorithm in an extension region are different from the corresponding part in the neighboring segment. Therefore, the final results reconstructed by the no inter-communication model differ from the sequential (non-parallel) algorithm.

However, research has shown that there is no significant image quality difference between the reconstructed 3D images of the no inter-communication model and the non-parallel reconstruction [14]. All major tissue features and structural information can be observed clearly.

Because two extension regions need to be computed with each 3D segment, this model increases the amount of computational workload per process. The extension regions are overlapped by neighboring segments, and redundant computation is introduced. This added computation can introduce some performance loss.

2.3 Overlap with inter-communication model

The second model reduces the amount of redundant computation by copying the data from corresponding

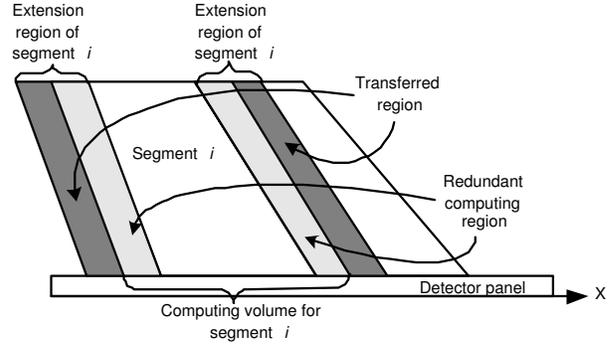


Figure 4. Overlap with inter-communication model.

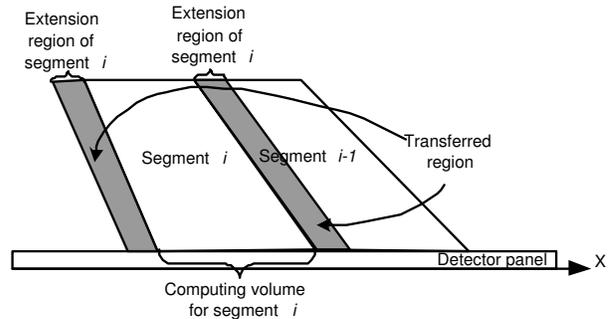


Figure 5. Non-overlap model.

neighboring segments (see Figure 4). Each segment obtains part of its extension region from the corresponding neighboring segment after each iteration of the reconstruction is completed. This method helps to keep the rest of the extension region computed locally.

The overlap in the inter-communication model does not have to compute the complete extension region, however, it introduces communication and synchronization overheads. After forward and backward projection computations are completed, each segment sends and receives part of the updated data to and from its neighbors. The advantage of this model is that the size of data exchanged between neighboring segments can be adjusted to adapt to the network speed. This model allows a user to optimize the performance on different systems by changing the amount of data transferred in the extension region.

2.4 Non-overlap model

The third model eliminates all redundant computation for computing the extension regions. All data in both extension regions is transferred from corresponding segments (see Figure 5). As with the overlap with inter-communication model, the non-overlap model exchanges reconstruction information between neighboring segments after the forward and backward projection computation phases. The difference is that the non-overlap model exchanges the entire neighboring region, and so computation can be performed locally.

The non-overlap model does not have reconstruction overlapped. As a result, the model has the highest communication overheads. The amount of data exchanged between neighboring segments depends on the thickness of the object and the geometry of the partitioning. The further from the chest wall the data represents, more data is transferred (see Figure 2). Generally, over 10MBs of data is transferred between two neighboring segments. Unlike the first two models, the non-overlap model produces the same image as the serial implementation.

In summary, the no inter-communication model is a coarse-grained parallelization approach. It requires less implementation effort, since there is no inter-processor communication performed during reconstruction. Furthermore, its performance will not rely on a high-performance network.

However, since redundant computation is wasteful the performance of this implementation depends on the CPU power. The non-overlap model, on the other hand, eliminates all redundant computation while requiring communication and synchronization between all neighboring segments. Its performance depends on both the CPU speed and on the network bandwidth/latency.

In contrast to the first model, the non-overlapped model uses a fine-grained parallelization approach and is much more difficult to implement. The overlap with inter-communication model can provide a better solution, for both the non inter-communication models and the non-overlap model fail to achieve the best performance.

3 Experiments and performance analysis

To find the best parallel implementation of ML-based reconstruction of tomosynthesis of breast cancer images, we developed a parallel version of the tomosynthesis reconstruction. The parallelization of tomosynthesis is implemented using in Message Passing Interface (MPI) 1.0 [1]. The parallel code is based on a Visual C++ pro-

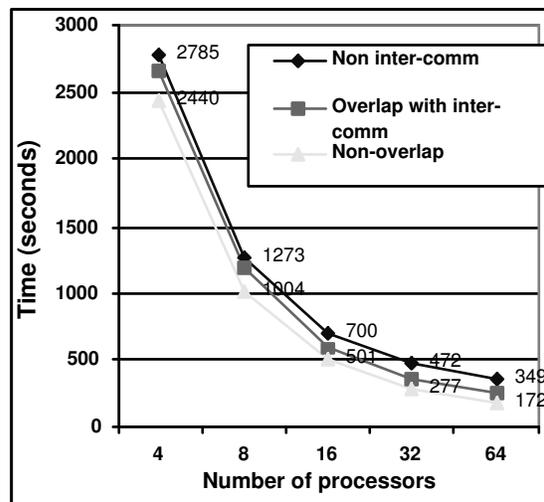


Figure 6. Performance of three implementations on number of processors, using UIUC NCSA’s Titan cluster.

gram developed at MGH. The program is compiled on two different operating system environments, Linux and UNIX (IBM AIX), and is executed on five different parallel systems shown in Table 1. The platforms tested in our experiments are comprised of a low-end PentiumIV cluster to high-end supercomputers of shared memory architecture.

A phantom dataset used in Tomosynthesis quality control is chosen in our performance tests. The size of our image is 1600×2304 pixels. The size of our 3D volume is $1600 \times 2304 \times 45$ pixels, where thickness of the phantom is recorded in 45mm. The size of an extension of a segment varies from 10 to 20, depending on the distance from the chest wall to the location of the partitioned segment. Based on the experiments from clinical trials, the parallelized ML algorithm completes in 8 iterations. The length of the extension region in the X-axial direction is pre-calculated, and varies from 15 to 20 pixels.

3.1 Performance Evaluation

Figure 6 shows runtime results of our model’s performance when given 4, 8, 16, 32 and 64 processors on the NCSA Titan cluster at UIUC. We observed very similar speedup trends for all three models. The best performance was obtained by the non-overlap model running on 64-processor.

	Processor	Interconnection
Intel P4 Cluster	2.5 GHz Pentium 4	100 Mb/s Ethernet
UIUC NCSA IBM p690	1.3 GHz POWER 4	Gbit Ethernet shared memory system
UIUC NCSA Intel Titan cluster	800 MHz Itanium-1 dual-processor	Gbit Myrinet shared L3 cache
SGI Altix 3300 shared memory system	1.3 GHz Itanium-2 dual-processor	NUMA-link interconnect shared memory system
U. of Michigan CAC Hypnos cluster	1.7 GHz Athlon 2000MP dual-processor	Gbit Myrinet

Table 1. Processor and interconnection network specifications of testing platforms.

In most of our tests, the non-overlap method is able to outperform the other two models by 200 to 300 seconds. When running on 64 processors, the fast reconstruction process using the non-overlap model only takes half of execution time when we used the no inter-communication model. These tests have demonstrated that the tomosynthesis reconstruction process is computationally-intensive, and that redundant computation increases this problem.

The results also demonstrate that the performance of the ML algorithm is very sensitive to the size of the allocated memory, even though its computational complexity grows linearly with the size of the 3D volume. In Figure 6, when the number of processes increases from 4 to 8, the size of each segment decreases by half, yet all three models can achieve more than two times speedup.

To understand the behavior of these three computing models, we have developed a fully instrumented parallel implementation in order to capture timing information of specified events during the reconstruction process, including forward projection, backward projection, synchronization time, communication time, segments collecting and file IO.

Figure 7 illustrates the cumulative time of the above events from the 8-iteration reconstruction process as run on the NCSA Titan cluster at UIUC. Data presented in Figure 7 is recorded by the root process (process rank is 0) of total 32 processes.

It is clear that the computational work of all three methods dominates overall performance. Experiments of the non-overlap method show that evenly partitioned segments lead to well-balanced workloads. However, when extension regions are added to the segment reconstruction, the overlap with inter-communication model and the non inter-communication model show increased synchronization time. The amount of increased synchronization is caused by the imbalance of the workloads between processes. The master process is assigned the

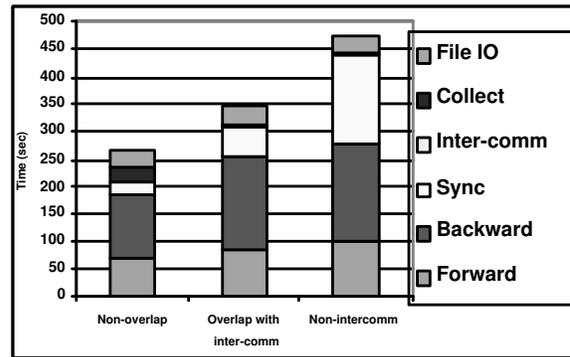


Figure 7. Parallelization methods comparison, using 32 process on the NCSA Titan cluster at UIUC.

partition next to the chest-wall which has the smallest extension region as compared to other partitions.

Figure 8 shows the performance of the non-overlap model executed on five different platforms using 32 processors. The 64-bit architecture of the Intel Itanium2 processor is on the SGI system and introduces the least computation and communication overhead. A Pentium4 cluster was installed with a 100 Mb/s ethernet switch, and experiences the largest communication overhead.

The forward and backward projection results illustrate the computing power of five different CPU architectures. Our test results show that the Itanium2 is able to run two times faster than the Pentium4 processor.

Figure 9 compares execution time of the reconstructions on five different platforms, all running on 32 processors. The non-overlap model has achieved the best performance among the three parallelization models, and this is true on all five different systems. Pentium4 cluster suffers significant performance loss when the ex-

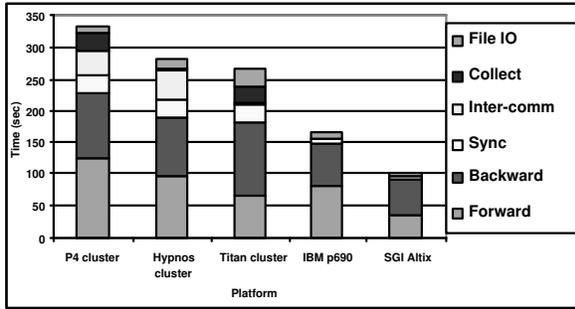


Figure 8. System comparison using the non-overlap implementation, running on 32 processors.

tension region is included. This is due to imbalance between processes. Further investigation is required to further fine-tune the workloads on each process.

In Figure 6, we can find that the fine-grained parallelization is two times faster than the coarse-grained one, as run on 64 processors. However, the performance difference between the three methods is barely noticeable on the SGI Altix platform (see Figure 9). For the SGI Altix system, the computing of an extension segment requires as much time as transferring the data from neighboring segment.

The SGI Altix system has outperformed all other systems in our experiments. However, its performance-cost ratio may not be the most attractive. Because the current tomosynthesis system is still a prototype, this work provides options for the most cost-effective system design.

4 Related work

Parallel and distributed computing are commonly used to solve computationally intensive problems in a number of areas. Parallel computing presently plays an important role in the developing area of biomedical image understanding, where large volumes of data need to be processed in a short amount of time.

General parallelization methods can be found in the literature [6, 2]. Parallel image rendering techniques studied by researchers are only capable of exploiting limited parallelism [9, 7].

In [8], a system is described that can process images over 100 times larger used by tomosynthesis. Reconstruction algorithms used by computerized tomography (CT) are not able to produce 3D high-resolution images

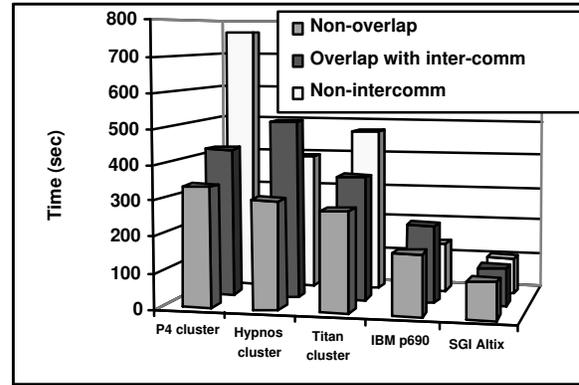


Figure 9. Performance test of 3 parallelization models on all platforms using 32 processors.

for practical use. There are some recent projects looking at parallelizing an ordered subset convex algorithm (OSC) on a shared memory computer [4]. However, its implementation is not portable to tomosynthesis and it relies heavily on a specified architecture to obtain the desired performance

Wu has introduced a coarse-grained parallelization approach in tomosynthesis reconstruction process [14]. 3D volume rendering of an object and its projection images are divided into multiple partitions, and part of the region on the boundary of a partition is overlapped by its neighboring partition. Partitions are reconstructed independently on distributed workstations, and then re-assembled together at the end. Wu's approach can not produce the exact reconstruction results as the sequential (non-parallel) algorithm, however, no significant visual difference have been found in clinical trials. This research targets a more efficient and generalizable parallelization strategy, whose implementation can be ported to different computing platforms easily. Our method can also produce identical results as the sequential algorithm at the expense of intensive inter-process communication. Performance differences between the coarse-grained and the fine-grained parallel method are analyzed.

5 Conclusion and Future Work

Tomosynthesis mammography has been shown to be a very effective method in the detection and diagnosis of cancer in breast tissue. However, during clinical trials, compute-time overhead has been identified

a major obstacle for doctors and physicians to use tomosynthesis effectively. Based on our analysis of the image reconstruction algorithms used in tomosynthesis, we have evaluated three partitioning models: no inter-communication, overlap with inter-communication and non-overlap model, resembling coarse-grained to fine-grained parallelization approach. The key tradeoff evaluated in these three models is the balance between computation and communication.

These models are designed to adapt different parallel and distributed systems with varied computational power and network speed. We tested our implementations on five parallel and distributed systems, including a low-cost Pentium4 cluster and a high-end IBM P690 supercomputer.

The test results show that the non-overlap model outperforms the other two in all test environments, though it has the largest amount of data exchanged between processes. Our experiments have demonstrated that using computation to replace communication may increase overall performance. The results of this research work have provided MGH with crucial information to develop the next generation of tomosynthesis products.

Future research work on tomosynthesis currently relies on the implementation of the non-overlap model to reconstruct 3D breast images. Because tomosynthesis provides 3D structural information of the breast, the reconstruction results can be used to guide the breast tumor biopsy. A computer-guided breast tumor biopsy requires real-time or close to real-time image reconstruction, so that doctors can accurately locate the exact position of a probe or a suspicious tumor tissue inside a breast. In our experiments, the best performance obtained is under 1 minute on a 64-processor SGI Altix system by using the non-overlap method, when this object is set in $1600 \times 2304 \times 45$. However, such performance can not be tolerated in biopsy. The reconstruction algorithm needs to be simplified and the image resolution could be reduced as well. Further investigation is required, however, current results will guide our future research work.

6 Acknowledgements

This work was supported by CenSSIS, the Center for Subsurface Sensing and Imaging Systems, under the Engineering Research Centers Program of the NSF (Award Number EEC-9986821), and by and the NSF Major Research Instrumentation Program (Award Number MRI-9871022).

References

- [1] Message Passing Interface Forum. MPI: A message-passing interface standard. 2003.
- [2] I.T. Foster. *Designing and Building Parallel Programs*. Addison-Wesley Publishing Company, 1995.
- [3] D. G. Grant. Tomosynthesis: A three-dimensional radiographic imaging technique. *IEEE Trans. Biomed. Eng.*, 19, 1972.
- [4] J. S. Kole and F. J. Beekman. Parallel statistical image reconstruction for cone-beam x-ray ct on a shared memory computation platform. *Physics in Medicine and Biology*, 50(6):1265–1272, 2005.
- [5] D.B. Kopans. *Breast Imaging 2nd ed.* Lippincott Williams and Wilkins, 1997.
- [6] V. Kumar, A. Grama, A. Gupta, and G. Karypis. *Introduction to Parallel Computing - Design and Analysis of Algorithms*. Benjamin/Cummings Publishing Company, 1994.
- [7] C.S. Leo and H. Schroder. Fast processing of medical images using a new parallel architecture, the hybrid system. In *Image Analysis and Interpretation, 2002. Proceedings. Fifth IEEE Southwest Symposium on*, pages 148–152, 2002.
- [8] V. Messerli, B. Gennart, and R.D. Hersch. Performances of the ps2 parallel storage and processing system for tomographic image visualization. In *Parallel and Distributed Systems, 1997. Proceedings.*, pages 514–522, 1997.
- [9] U. Neumann. Interactive volume rendering on a multi-computer. In *Proceedings of the ACM 1992 Symposium on Interactive 3D Graphics*, pages 87–93, 1992.
- [10] L. T. Niklason, B. T. Christian, L. E. Niklason, D. B. Kopans, D. E. Castleberry, B. H. Opsahl-Ong, C. E. Landberg, P. J. Slanetz, A. A. Giardino, R. M. Moore, D. Albagi, M. C. DeJule, P. A. Fitzgerald, D. F. Fobare, B. W. Giambattista, R. F. Kwasnick, J. Liu, S. J. Lubowski, G. E. Possin, J. F. Richotte, C-Y Wei, and R. F. Wirth. Digital tomosynthesis in breast imaging, 1997.
- [11] S. Suryanarayanan, A. Karellas, S. Vedantham, S. P. Baker, S. J. Glick, C. J. D’Orsi, and

R. L. Webber. Evaluation of linear and nonlinear tomosynthetic reconstruction methods in digital mammography. *Acad. Radiol.*, 8, 2001.

- [12] S. Suryanarayanan, A. Karellas, S. Vedantham, S. J. Glick, C. J. D'Orsi, S. P. Baker, and R. L. Webber. Comparison of tomosynthesis methods used with digital mammography. *Acad. Radiol.*, 7, 2000.
- [13] A. Stewart T. Wu, M. Stanton, T. McCauley, W. Phillips, D. B. Kopans, R. H. Moore, J. W. Eberhard, B. Opsahl-Ong, L. Niklason, and M. B. Williams. Tomographic mammography using a limited number of low-dose cone-beam projection images. *Med. Phys.*, 30(3), 2003.
- [14] T. Wu, J. Zhang, R. Moore, E. Rafferty, D.B. Kopans, W. Meleis, and D. Kaeli. Digital tomosynthesis mammography using a parallel maximum-likelihood reconstruction method. volume 5368, pages 1–11. SPIE, 2004.