

Performance Study of Nonrigid Registration Algorithm for Investigating Lung Disease on Clusters*

Fumihiko Ino¹ Yuya Tanaka¹ Hiroko Kitaoka² Kenichi Hagihara¹

¹Graduate School of Information Science and Technology, Osaka University
1-3 Machikaneyama, Toyonaka, Osaka 560-8531, Japan

²Graduate School of Medicine, Osaka University
2-2 Yamadaoka, Suita, Osaka 565-0871, Japan

E-mail: ino@ist.osaka-u.ac.jp

Abstract

This paper presents a performance study of a nonrigid registration algorithm for investigating lung disease on clusters. Our algorithm combines two conventional acceleration techniques in order to achieve fast registration: a data-parallel processing technique for accelerating the registration procedure; and a precomputation technique for reducing the computational complexity. We perform some experiments on three clusters with different CPU and network performance in order to make clear what kinds of acceleration techniques and computing environments provide higher performance. The results show that a cluster with Gigabit Ethernet (GbE) network is the most cost effective solution that reduces registration time from ten hours to ten minutes with a linear speedup.

Keywords: image registration, medical image processing, cluster computing, MPI, performance evaluation.

1. Introduction

Image registration [1] is a technique for establishing point-to-point correspondences between different images taken at different time, from different viewing points, in different modalities. This technique assists medical doctors in various diagnoses. For example, it helps them in detecting cancers by monitoring changes in size, shape, or image intensity over time intervals.

One problem with this technique is that it requires time-consuming computations in order to realize accurate, robust, and completely automated alignments. For example, it takes several hours to align three-dimensional (3-D) clinical images obtained by modern imaging systems such as

*This work was partly supported by JSPS Grant-in-Aid for Scientific Research on Priority Areas (17032007).

X-ray computed tomography (CT) scans [8]. Therefore, some acceleration techniques are needed to utilize registration during surgery, where registration time is strictly restricted within ten minutes to minimize the patient's strain.

There are many papers [2, 3, 8, 10–12] reporting experiences in using parallel machines to make registration time short enough to perform intraoperative registration. These prior projects successfully accelerate the registration procedure by reducing computational complexity and by parallelizing performance bottlenecks in the procedure. Although they present that combining some acceleration techniques realizes intraoperative registration, it is still not clear how efficiently each technique reduces registration time.

In this paper, we study the performance of a nonrigid registration algorithm, aiming at making clear what kinds of acceleration techniques and computing environments are effective to reduce registration time. To do this, we implement a registration algorithm for investigating lung disease on clusters. Our algorithm is based on an optimization approach [9], which resolves the registration problem into an optimization problem. We combine this approach with two conventional acceleration techniques: a data-parallel processing technique for accelerating the registration procedure; and a precomputation technique for reducing the computational complexity.

The remainder of the paper is as follows. Section 2 presents a brief overview of the registration algorithm with the two acceleration techniques mentioned above. Section 3 shows the performance study on three clusters of PCs. Finally, Section 4 summarizes the paper.

2. Nonrigid Registration

Let F and R be the floating and reference images, respectively. The nonrigid registration problem is defined as computing a nonrigid transformation T that aligns F to R .

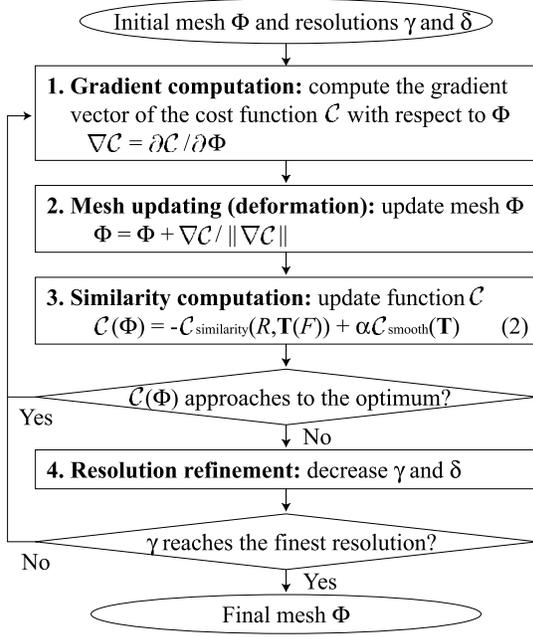


Figure 1. Optimization-based registration algorithm [5]. It minimizes the cost function \mathcal{C} by the steepest descent optimization [7] in a coarse-to-fine manner. The final mesh Φ determines the final transformation \mathbf{T} .

To compute this, our algorithm uses an optimization approach [5], which aligns images through the optimization of a cost function \mathcal{C} associated with a similarity measure between the images (see Figure 1). This optimization is performed by means of the steepest descent optimization [7] in a coarse-to-fine manner. This hierarchical alignment strategy contributes to reduce the computational complexity.

Our algorithm has four advantages as follows:

- Hierarchical, locally controlled deformations [9] by B-spline functions [4];
- Robust similarity measure by information theory [5];
- Fast B-spline interpolation by precomputation [8];
- Fast optimization by parallelization [11].

2.1. Hierarchical Deformation Model

Our algorithm employs B-spline functions in order to represent the nonrigid transformation \mathbf{T} at lower complexity. B-spline functions are computationally efficient because they use interpolation to realize hierarchical, locally controlled free-form deformations (FFDs).

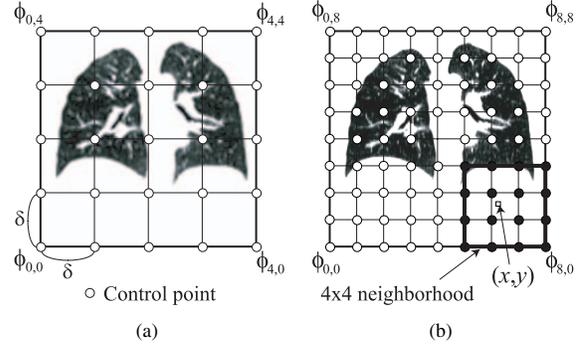


Figure 2. Hierarchical B-spline free-form deformations (FFDs) [9]. (a) Deformations of the floating image are performed by manipulating an overlaying mesh of control points in a coarse-to-fine manner. δ represents the spatial resolution of control points and $\phi_{i,j}$ represents a control point. (b) The deformation of point (x,y) is determined by its surrounding 4×4 neighborhood of control points.

Figure 2 shows how this interpolation is applied to image F . The nonrigid transformation \mathbf{T} is given by manipulating a mesh of control points overlaid in the image domain $\Omega = \{(x, y, z) \mid 0 \leq x < X, 0 \leq y < Y, 0 \leq z < Z\}$. Given a mesh Φ of control points $\phi_{i,j,k}$, the transformation \mathbf{T} of point (x, y, z) in image F is defined by

$$\mathbf{T}(x, y, z) = \sum_{l=0}^3 \sum_{m=0}^3 \sum_{n=0}^3 B_l(u) B_m(v) B_n(w) \phi_{i+l, j+m, k+n}, \quad (1)$$

where $u = x/\delta - \lfloor x/\delta \rfloor$, $v = y/\delta - \lfloor y/\delta \rfloor$, $w = z/\delta - \lfloor z/\delta \rfloor$, $i = \lfloor x/\delta \rfloor - 1$, $j = \lfloor y/\delta \rfloor - 1$, $k = \lfloor z/\delta \rfloor - 1$, and B_l represents the l -th basis function of cubic B-splines. This equation means that the deformation of any point (x, y, z) is given by interpolating the deformations of its surrounding $4 \times 4 \times 4$ neighborhood of control points (see Figure 2(b)).

Recall here that our algorithm aligns images in a coarse-to-fine manner. To do this, the algorithm organizes mesh Φ , images F and R in a hierarchy. Then, it decreases the spatial resolutions of images and control points, γ and δ , respectively, at each level of hierarchy.

2.2. Cost Function

The cost function for optimization is defined by Rueckert et al. [9] as follows:

$$\mathcal{C}(\Phi) = -\mathcal{C}_{\text{similarity}}(R, \mathbf{T}(F)) + \alpha \mathcal{C}_{\text{smooth}}(\mathbf{T}), \quad (2)$$

where $C_{\text{similarity}}(R, \mathbf{T}(F))$ is the similarity measure between the reference image R and the transformed floating image $\mathbf{T}(F)$, C_{smooth} is the penalty term that constrains the spline-based transformation to be smooth, and α is the weighting parameter which is determined experimentally.

For the similarity measure $C_{\text{similarity}}$, our algorithm employs a general measure based on information theory. This information-based strategy does not depend on specific features of images, for example, contours and surfaces, so that it realizes robust registration of multimodality images [5, 8, 9]. Although there are many measures such as mutual information [5] and its normalized version [9], we have experimentally determined to use correlation coefficient for lung registration. In summary, the similarity measure between images A and B is given by

$$C_{\text{similarity}}(A, B) = \frac{\sum(A(x, y, z) - \bar{A})(B(x, y, z) - \bar{B})}{\sqrt{\sum(A(x, y, z) - \bar{A})^2 \sum(B(x, y, z) - \bar{B})^2}}, \quad (3)$$

where $A(x, y, z)$ is the image intensity at point (x, y, z) in image A , and \bar{A} is the mean intensity in image A .

The penalty term C_{smooth} intends to regularize the transformation in order to avoid unsmooth deformations. For lung registration, we use the following constraint model:

$$C_{\text{smooth}} = \frac{1}{V} \int_0^X \int_0^Y \int_0^Z \left[\left(\frac{\partial^2 \mathbf{T}}{\partial x^2} \right)^2 + \left(\frac{\partial^2 \mathbf{T}}{\partial y^2} \right)^2 + \left(\frac{\partial^2 \mathbf{T}}{\partial z^2} \right)^2 + 2 \left(\frac{\partial^2 \mathbf{T}}{\partial xy} \right)^2 + 2 \left(\frac{\partial^2 \mathbf{T}}{\partial xz} \right)^2 + 2 \left(\frac{\partial^2 \mathbf{T}}{\partial yz} \right)^2 + \frac{\lambda}{2} \left(\frac{\partial^2 \mathbf{T}}{\partial x^2} + \frac{\partial^2 \mathbf{T}}{\partial y^2} + \frac{\partial^2 \mathbf{T}}{\partial z^2} \right)^2 \right] dx dy dz, \quad (4)$$

where V denotes the volume of the image domain Ω , and λ represents the elastic coefficient for lung. From the viewpoint of physics, this term approximates the energy of an isotropic elastic object (namely lung tissue) which is subjected to deformations with volume change.

As we mentioned before, our algorithm employs the steepest descent optimization [7] to find the optimal transformation parameter Φ that minimizes Eq. (2). The algorithm stops this optimization if a local optimum has been found. Here, it assumes a local optimum if $\|\nabla \mathcal{C}\| \leq \epsilon$, where $\nabla \mathcal{C} = \partial \mathcal{C} / \partial \Phi$, and $\|\nabla \mathcal{C}\|$ and ϵ represent the gradient norm of the cost function \mathcal{C} and a threshold for minimization, respectively. The gradient $\nabla \mathcal{C}$ is estimated by using the finite-difference approximation [7].

2.3. Precomputation for B-spline Interpolation

In order to reduce the computational complexity of B-spline FFDs, we applied Rohlfing's precomputation tech-

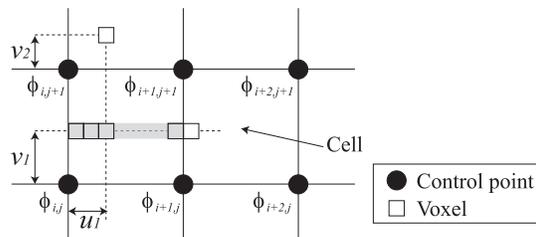


Figure 3. Precomputation [8] for B-spline interpolation. Gray colored voxels have the same value $\hat{\phi}_{i+l}$, and thereby this term should be computed only once.

nique [8] to the algorithm. The key idea of this technique is precomputation that intends to eliminate redundant computations which are repeatedly for some different points.

Suppose that we have a zyx -loop to compute $\mathbf{T}(x, y, z)$ for all points (x, y, z) in the image domain Ω . Eq. (1) can be rewritten as $\mathbf{T}(x, y, z) = \sum_{l=0}^3 B_l(u) \hat{\phi}_{i+l}$, where $\hat{\phi}_{i+l} = \sum_{m=0}^3 \sum_{n=0}^3 B_m(v) B_n(w) \phi_{i+l, j+m, k+n}$. Then, term $\hat{\phi}_{i+l}$ keeps the same value for all points (x, y, z) in one row located within the same cell of the mesh (see Figure 3). Therefore, placing this computation outside the x -loop reduces the total computational cost for the zyx -loop.

This modification allows us to compute the basic B-spline function only $4n^3 + 4^2n^2 + 4^3n$ times while the unoptimized implementation computes it $192n^3$ times, where n represents the number of points in the image. Therefore, the execution time for object deformations will be significantly reduced by 98%.

2.4. Parallelization

Although the precomputation technique contributes to reduce registration time, parallelization is still required to make it be compatible with surgical usage.

Because nonrigid registration is not a data-intensive application, we assume that all computing nodes have the entire images in their local memory. Actually, a $512 \times 512 \times 512$ voxel image consisting of 2-bytes requires 256MB of memory, so that this assumption is reasonable for modern PCs equipped with more than 512MB of main memory.

We first analyzed our sequential implementation to locate performance bottlenecks:

- Gradient computation. This computation phase is required for the steepest descent optimization of the cost function. It takes approximately 57% of total registration time.
- Similarity computation. Computing the cost function

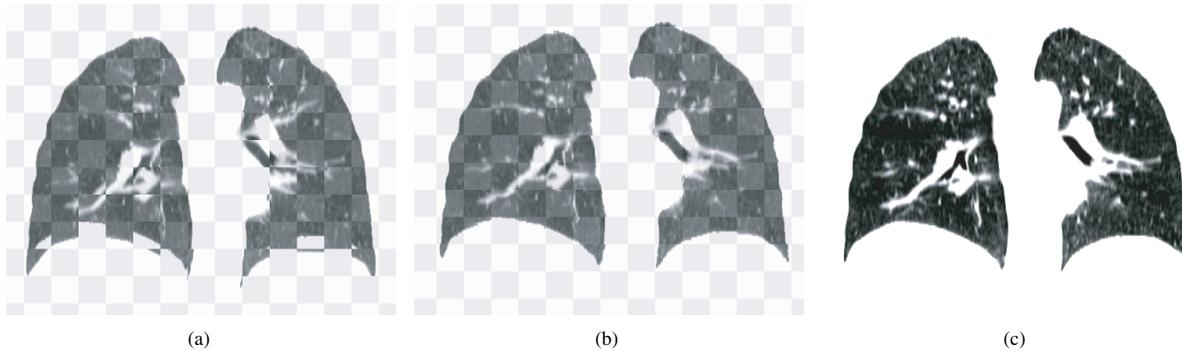


Figure 4. Checkerboard visualization of reference and floating images (a) before and (b) after registration. The registered image in (b) has the same contours as in (c) the reference image.

\mathcal{C} is also a performance bottleneck. It takes 42% of the total time.

- **Image sampling.** This phase occurs when the algorithm increases the hierarchy of alignment. Although this phase is a small bottleneck on single CPU machines, it should be parallelized on parallel machines, because I/O operations usually become a larger bottleneck as the number of CPUs increases.

Each of the above computation phases has parallelism in the image space. Therefore, we can parallelize each phase by decomposing the image domain into small portions and assigning them to CPUs. After processing these portions in a data-parallel manner, each CPU has a local computation result, so that a collective communication is needed to reduce local results into a global result. Thus, the entire workload can be decomposed into many independent pieces.

For this workload distribution, our parallel algorithm employs a cyclic distribution, because we assume no data decomposition. This cyclic strategy statically balances workloads among CPUs. In other words, static load balancing is sufficient for our algorithm whose workload is generated uniformly in the image domain.

3. Performance Study

To make clear the timing benefits of the acceleration techniques mentioned above, we now study the performance of the registration algorithm using three Linux clusters with different performance characteristics in terms of CPU and network. Our registration algorithm is implemented using the C++ language and the Message Passing Interface (MPI) standard [6].

Table 1. Dataset specification.

Level	Image size	Number of control points
1	$33 \times 33 \times 36$	252
2	$65 \times 65 \times 70$	1,848
3	$129 \times 129 \times 138$	13,041
4	$257 \times 257 \times 275$	97,785

We performed nonrigid registration using a pair of lung CT images (see Figure 4). Table 1 shows its specification with the hierarchical organization. The file size of an image is 157MB.

Table 2 summarizes our clusters. Each cluster has two interconnection networks: one is specialized for data communication between computing nodes; and the other, the Ethernet-based network, is basically for I/O from/to the centralized storage server. Note here that cluster #1 is a uni-processor (UP) system while the remaining clusters are 2-way symmetric multiprocessor (SMP) systems.

3.1. Acceleration Results by Precomputation

Table 3 shows the timing results measured on a single node in the clusters. The precomputation technique reduces registration time by 21–39%. These results are reliable, because the execution time for object deformations, which is included in registration time, is reduced by 83–86% while our theoretical analysis predicts a reduction of at most 98%. Thus, measured reduction is close to predicted reduction.

These timing results also indicate that sufficient tunings are essential to parallelize sequential implementations in a proper manner. For example, if the precomputation technique was not applied to the sequential implementation, we might have parallelized the code for B-spline interpolation.

Table 2. Cluster specification. Network bandwidth and latency are measured using a performance measurement tool (<http://www.scl.ameslab.gov/netpipe/>).

Component	Cluster #1	Cluster #2	Cluster #3
CPU	Xeon 3.2GHz	Xeon 2.8GHz	Pentium III 1GHz
Cache size	512KB	512KB	256KB
Number of nodes	32	16	64
Number of CPUs per node	1	2	2
Network #1	InfiniBand (InB)	Myrinet-PCIX (MYX)	Myrinet-2k (MY)
Bandwidth	3999Mb/s	1848Mb/s	1433Mb/s
Latency	10 μ s	6 μ s	9 μ s
Network #2	GbE	GbE	Fast Ethernet (FE)
Bandwidth	852Mb/s	897Mb/s	89Mb/s
Latency	22 μ s	17 μ s	48 μ s

Table 3. Precomputation results on 1 CPU.

Cluster	Original time (s)	Optimized time (s)	Reduction rate (%)
#1	18,352	11,209	39
#2	21,796	13,536	38
#3	59,424	46,997	21

However, this parallelization does not make sense if the code becomes a trivial bottleneck after applying precomputation, as we presented in Table 3.

3.2. Acceleration Results by Parallelization

Table 4 shows the parallelization results on three clusters. Note here that we basically measure the performance on a UP configuration. SMP configurations are employed only if all CPUs are required to measure the performance.

Our implementation successfully reduces registration time as the number of CPUs increases, so that registration time is reduced from hours to minutes. As a result, it achieves linear speedups in most cases. Here, the speedup S_p on p CPUs is given by $S_p = T_1/T_p$, where T_p represents registration time on p CPUs. In this table, there are four important points to be mentioned.

- GbE is most cost effective network in terms of performance. Although some special networks such as InfiniBand (InB) and Myrinet-PCIX (MYX) provide higher bandwidth with shorter latency (see Table 2), there is no significant difference with respect to the performance.
- When using less than 8 nodes with MYX (cluster #2), the speedup exceeds the number of CPUs. This superlinear speedup is due to the memory hierarchy, because our parallel scheme allows CPUs to access a

smaller part of the image domain, as compared to the sequential implementation. Therefore, as the number of CPUs increases, we have higher cache utilization. Note here that, at the same time, CPUs generally spend longer time for communication. Thus, the superlinear speedup appears only when the program achieves higher cache utilization with less communication.

- When using more than 32 nodes with Fast Ethernet (FE) network (cluster #3), we obtain lower speedups. This lower acceleration is due to its lower network bandwidth, because it causes network contention. This can be explained by Table 5, which presents the ratio of communication time. Almost half of the entire time is spent for communication, mainly due to image loading from the storage server.
- When using more than 32 nodes with Myrinet (MY) network (cluster #3), we also obtain lower speedups. In contrast to FE network, the communication ratio on MY network is kept a lower value even when using 128 nodes (Table 5). Therefore, this lower acceleration is due to computation rather than communication. The reason for this is that SMP systems, which share the memory bus with CPUs in a node, have lower memory bandwidth, as compared to UP systems. Table 6 presents how this lower bandwidth drops the registration performance. In this table, we can see that the parallel efficiency for a SMP configuration is 10% lower than that for UP configurations, though all the three configurations use the same number of CPUs. Thus, when using 128 CPUs on cluster #3, it takes longer time to fetch the data from the main memory, as compared with UP configurations. In summary, the SMP architecture decreases the parallel efficiency for memory-intensive registration applications.

Table 4. Parallelization results on p CPUs. Times are presented in seconds.

Cluster	Registration time (speedup)							
	$p = 1$	$p = 2$	$p = 4$	$p = 8$	$p = 16$	$p = 32$	$p = 64$	$p = 128$
#1 w/ InB	11,209 (1.0)	5,880 (1.9)	2,811 (4.0)	1,491 (7.5)	767 (15)	413 (27)	—	—
#1 w/ GbE	11,209 (1.0)	5,682 (1.9)	2,843 (3.9)	1,513 (7.4)	811 (14)	457 (25)	—	—
#2 w/ MYX	13,536 (1.0)	6,617 (2.0)	3,423 (4.0)	1,681 (8.1)	872 (16)	507 (27)	—	—
#2 w/ GbE	13,536 (1.0)	7,107 (1.9)	3,685 (3.7)	1,692 (8.0)	889 (15)	539 (25)	—	—
#3 w/ MY	46,997 (1.0)	24,896 (1.9)	12,556 (3.7)	6,369 (7.4)	3,250 (14)	1,756 (27)	965 (49)	760 (62)
#3 w/ FE	46,997 (1.0)	24,915 (1.9)	12,793 (3.7)	6,726 (7.0)	3,705 (13)	2,332 (20)	1,661 (28)	1,547 (30)

Table 5. Communication-to-computation ratio presented in percentages.

Cluster	p : number of CPUs							
	2	4	8	16	32	64	128	
#1 w/ InB	0.02	0.1	0.2	0.5	0.8	—	—	
#1 w/ GbE	0.1	0.5	2.2	5.6	10.8	—	—	
#2 w/ MYX	0.04	0.1	0.3	0.6	1.6	—	—	
#2 w/ GbE	0.07	0.2	0.6	1.2	3.7	—	—	
#3 w/ MY	0.02	0.1	0.1	0.3	0.9	2.4	2.6	
#3 w/ FE	0.4	1.6	5.0	12.7	25.4	42.8	52.0	

Table 6. Performance comparison between UP and SMP systems running on a 2-CPU configuration (cluster #3). Parallel efficiency is given by the speedup divided by 2 CPUs.

Breakdown	SMP	UP w/ MY	UP w/ FE
Image loading (s)	34	33	113
Gradient Computation (s)	16,072	14,052	13,926
Similarity Computation (s)	11,598	10,503	10,554
Others (s)	314	308	322
Total (s)	28,018	24,896	24,915
Parallel efficiency (%)	84	94	94

4. Conclusions

We have presented a performance study of a nonrigid registration algorithm using three clusters. Our algorithm is based on an optimization approach with two conventional acceleration techniques: a data-parallel processing technique; and a precomputation technique.

The experimental results show that (1) the precomputation technique realizes efficient B-spline deformations and reduces registration time by 39%; (2) the data-parallel processing technique is necessary to reduce registration time from hours to minutes; (3) it also achieves a linear speedup on clusters with GbE network; and (4) faster I/O and network infrastructures are essential to achieve higher parallel efficiency for large-scale clusters with more than 32 nodes. We also have indicated that sequential implementations should be optimized well enough in advance of parallelization in order to avoid unnecessary parallelization.

References

- [1] J. V. Hajnal, D. L. Hill, and D. J. Hawkes, editors. *Medical Image Registration*. CRC Press, Boca Raton, FL, 2001.
- [2] S. Hastings, T. Kurc, S. Langella, U. Catalyurek, T. Pan, and J. Saltz. Image processing for the grid: A toolkit for building grid-enabled image processing applications. In *Proc. CC-Grid'03*, pages 36–43, May 2003.
- [3] F. Ino, K. Ooyama, and K. Hagihara. A data distributed parallel algorithm for nonrigid image registration. *Parallel Computing*, 31(1):19–43, Jan. 2005.
- [4] S. Lee, G. Wolberg, and S. Y. Shin. Scattered data interpolation with multilevel B-splines. *IEEE Trans. Visualization and Computer Graphics*, 3(3):228–244, July 1997.
- [5] F. Maes, A. Collignon, D. Vandermeulen, G. Marchal, and P. Suetens. Multimodality image registration by maximization of mutual information. *IEEE Trans. Medical Imaging*, 16(2):187–198, Apr. 1997.
- [6] Message Passing Interface Forum. MPI: A message-passing interface standard. *Int'l J. Supercomputer Applications and High Performance Computing*, 8(3/4):159–416, 1994.
- [7] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery. *NUMERICAL RECIPES in C: The Art of Scientific Computing*. Cambridge University Press, Cambridge, UK, 1988.
- [8] T. Rohlfing and C. R. Maurer. Nonrigid image registration in shared-memory multiprocessor environments with application to brains, breasts, and bees. *IEEE Trans. Information Technology in Biomedicine*, 7(1):16–25, Mar. 2003.
- [9] D. Rueckert, L. I. Sonoda, C. Hayes, D. L. G. Hill, M. O. Leach, and D. J. Hawkes. Nonrigid registration using free-form deformations: Application to breast MR images. *IEEE Trans. Medical Imaging*, 18(8):712–721, Aug. 1999.
- [10] R. Stefanescu, X. Pennec, and N. Ayache. Grid-enabled non-rigid registration of medical images. *Parallel Processing Letters*, 14(2):197–216, June 2004.
- [11] S. Warfield, F. Jolesz, and R. Kikinis. A high performance computing approach to the registration of medical imaging data. *Parallel Computing*, 24(9/10):1345–1368, Sept. 1998.
- [12] S. K. Warfield, M. Ferrant, X. Gallez, A. Nabavi, F. A. Jolesz, and R. Kikinis. Real-time biomechanical simulation of volumetric brain deformation for image guided neurosurgery. In *Proc. SC'00*, pages 1–16, Nov. 2000.