Proceedings of the 2007 IEEE Symposium on Computational
Intelligence in Image and Signal Processing (CIISP 2007)

High Performance Medical Image Registration Using a Distributed Blackboard Architecture

Roger J. Tait, Gerald Schaefer, Adrian A. Hopgood and Tomoharu Nakashima

***Abstract*—A major drawback of medical image registration techniques is the performance bottleneck associated with similarity computation. Such bottlenecks limit registration applications in situations where fast execution times are required. In this paper a novel framework for high performance intensity-based medical image registration is presented. Geometric alignment of both reference and sensed images is achieved through a combination of scaling, translation, and rotation. Crucially, similarity computation is performed intelligently by knowledge sources (KSs) organised in a worker/manager model. The KSs work in parallel and communicate with each other by means of a distributed blackboard architecture. Partitioning of the blackboard is used to balance communication and processing workloads. The registration framework presented demonstrates the flexibility of the coarse-grained parallelism employed and shows how high performance medical image registration can be achieved with non-specialised architectures. Experimental results obtained during testing show that substantial speedups can be achieved.**

I. Introduction

Transform parameter optimisation, re-sampling, and similarity computation form the basic steps of an intensity-based image registration algorithm [1][2]. During transform parameter optimisation, translation, rotation, and scaling parameters which geometrically map intensity co-ordinates in the reference (fixed) image to corresponding locations in the sensed (moving) image are estimated. Fixed image co-ordinates which map to non-integer locations require interpolation, this represents the re-sampling stage. Once re-sampled, a metric is used for similarity computation in which a degree of likeness between corresponding images is calculated [3]. Optimisation of the similarity measure is the goal of the registration process and is achieved by seeking the best transform parameters. Transform parameters are therefore defined as a search space. Importantly, due to the iterative nature of registration algorithms, computation of the similarity measure represents a performance bottleneck which limits the speed of time critical applications.

The similarity metric, used for generation of the similarity measure, works by examining corresponding intensities in both fixed and moving images and then formulating a measure based on the relationship between these intensities. The similarity metric is also required to assume that the relationship changes with variations in the spatial transformation used to map between images and a maximum measure of similarity is achieved when images are in close alignment. Selection of the metric is largely dependent on the type of registration problem to be solved [4]. For example, some metrics produce a search space with a large capture range that is well suited to the registration of images differing by large transformations. Other metrics, in contrast, are less computationally intensive and generate a search space that requires initial transform parameters to be close to optimum. Intensity equality, which is maximal when intensities are equal between images captured with the same sensor type, is one such relationship employed as a similarity metric in single-modal registration. Unfortunately, total equality is seldom reached due to noise and image acquisition inconsistencies.

To overcome the speed constraints associated with intensity-based medical image registration, high performance computing has been employed by a number of researchers. Clinically compatible speeds have been achieved by Warfield *et al*. [5] who introduced a parallel non-rigid algorithm based on the work-pile paradigm. In their research, a message passing interface and cluster of symmetric multi-processors execute parallelised similarity computation operations using POSIX threads. Results published by the group show that successful registration of brain scans has been achieved in less than 10 minutes. Christensen [6] in contrast compares two non-thread-based architectures, Multiple Instruction Multiple Data (MIMD) and Single Instruction Multiple Data (SIMD). The MIMD implementation is recorded as being four times faster than its SIMD counterpart. Reduced performance of the SIMD implementation is reportedly caused by overheads during serial portions of the algorithm.

More recently, multi-threaded programming together with data partitioning has been employed by Rohlfing *et al*. [7] to largely eliminate the need for explicit message passing between concurrent processes. Despite the need for specialised hardware, the scheme is reported to make implementation of high performance non-rigid registration a comparatively easy task when compared to other architectures. Using 64 CPUs registration of two $256 \times 256 \times 100$ voxel images was achieved in approximately 1.5 minutes. A similar data distributed parallel algorithm is

R.J. Tait and A.A. Hopgood are with the School of Computing and Informatics, Nottingham Trent University, Clifton Campus Nottingham, NG11 8NS, U.K., e-mail: roger.tait@students.ntu.ac.uk.

G. Schaefer is with the School of Engineering and Applied Sciences, Aston University, Birmingham, B4 7ET, U.K., e-mail: g.schaefer@aston.ac.uk.

T. Nakashima is with Prefecture University, Osaka, Japan.

described by Ino *et al*. [8]. Based on Schnabel's implementation the algorithm achieves efficient alignment using information theory and adaptive mesh refinement. Experimental results obtained on a 128 processor cluster show that images as large as 1024×1024×590 voxels can be aligned in minutes rather than hours. Importantly, the limitations of memory space when processing images is discussed in detail.

In this paper we introduce a novel parallel processing framework designed to achieve high performance intensity-based medical image registration. Unlike other implementations reported in the literature [9], the approach adopted is based a distributed blackboard architecture that supports multiple knowledge sources (KSs). Preconditions attached to KS rule files determine, in accordance with information on the blackboard, when a KS can make its contribution at any given moment. This reactive behaviour removes the need for a dedicated control module and associated KS management overheads. The modular architecture adopted is easily scalable and allows for addition of specialised KSs as required. Comparisons with non-distributed implementations confirm efficiency of the proposed approach.

II. The IMAGE Registration Framework

DARBS (Distributed Algorithmic and Rule-based Blackboard System), is a distributed blackboard architecture based on a client/server model [10], in which the server functions as a blackboard and client modules as KSs. The worker/manager model, based on pervious work with a distributed blackboard architecture [11], on which the framework is built is illustrated in Figure 1. Each KS shown represents a structure in which rules and algorithms can be embodied:

- The Distributor KS splits an image into segments which are then placed on the blackboard. The Distributor KS then terminates.
- Worker KSs take segments from the blackboard and perform local processing.
- The Manager KS is employed to co-ordinates Worker KS activities.

A. Partitioning of Framework Data

Storage on the blackboard of image data ensures equal access for all KSs. Division of the blackboard into partitions that correspond to KS types simplifies management of KS activities. Partitioning is also used to balance communication and processing workloads. The blackboard is divided into the following partitions:

- A *Distributor control* partition controls division of an image into segments.
- *Worker n control* partitions are used to manage processing of segments.
- Supervision of Worker KS activities is achieved by means of the *Manager control* partition.
- System variables are maintained in a *Parameters* partition.
- The *Image container* partition holds partitioned image segments.

Due to the exhaustive search required, a drop in performance can be expected with a single partition implementation. Similar inefficiency occurs through management and processing of excess partitions. Crucially, whenever the content of a partition is modified, the blackboard broadcasts a message informing all KSs that the partition has changed. Individual KSs then react to the changes depending on their implemented behaviour.

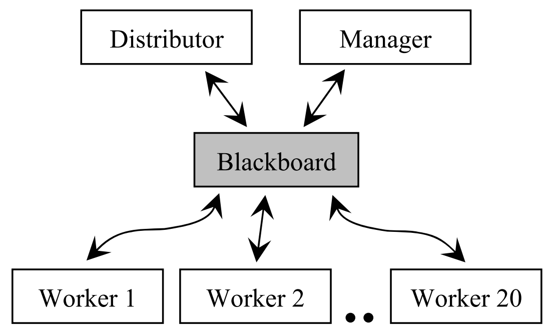


Fig 1. The worker/manager model on which the registration framework is based. Worker KSs perform concurrent processing of image segments while the Manager KS co-ordinates Worker KS activities.

B. Distributed Similarity Computation

In order for a registration algorithm to be distributed, both fixed and moving images require division into segments and distribution between Worker KSs. By employing gradient-based optimisation, the similarity metric can be used to produce derivatives of the similarity measure, with respects to each transform parameter. To achieve this, transform parameters require propagation to all Worker KS. On receiving the propagated parameters each Worker KS must compute local derivatives of the similarity measure, between the segments allocated to it. Once computed the local derivatives require accumulation and summation into a global derivative by the Manager KS. This allows transform parameters to be updated based upon the similarity between whole images. Convergence testing can then be performed, by the Manager KS, using the newly updated parameters. Depending on the success or failure of convergence testing, propagation of updated transform parameters and hence evaluation of the new parameters can occur.

C. Information Strings

To achieve distribution a range of strings were created to control firing of KS rules and allow the flow of transform and derivatives parameters between framework components. Example strings are shown in Figure 2. The region of interest string is used to hold the starting co-ordinates and size in corresponding dimensions, of a segment without borders. Region of interest strings are generated by the Distributor KS and placed in all worker control partitions. Created and updated by the Manager KS, the current parameters string is used for propagation of updated transform parameters to all Worker KSs. Generated by the Worker KSs, derivative strings are used for accumulation of local derivatives. Created by the Manager KS, the final parameters string contains optimal transform parameters.

```
[ROI 0_0_700_900]

[Current 1.34982342..._11.851]

[Derivative -203.6834..._54.901]

[Final 1.64514585..._15.79934]
```

Fig 2. Information strings for controlling the transform parameter optimisation process. Each string consists of an identifying tag followed by an underscore delimited list of numbers.

Employed as a trigger mechanism, the current parameter strings co-ordinate the activation of Worker KS activities. The creation of derivative strings, in contrast, marks the temporary suspension of Worker KS activities. Once generated the final parameters strings mark the permanent suspension of Worker KS activities.

III. KS BEHAVIOUR

Reading from and writing to the blackboard is implemented as standard functionality and provides a mechanism for communication between KSs. In the following section the basic KS behaviour, implemented as rule files, is described.

A. *The Distributor KS*

Initial tasks performed by the Distributor KS include clearance of all data from the blackboard. Selection of fixed and moving images is then manually performed after a simple viewer has been shown. Next, an initial transform is extracted from the selected images and formatted into a current parameters string. To extract the initial transform, centres of mass are computed for both fixed and moving images using moments of their intensity grey levels. The fixed image centre of mass is set as the rotational centre, while the translation component is set as the vector between the fixed and moving mass centres. Once created the current parameters string is added to the *Parameters* partition. Division of images into segments and sending to the *Image container* partition is then performed. Copies of the selected images are also sent to the *Image container* partition. Region of interest strings are generated for each segment and added to their associated worker control partition. Each region of interest generates a border at the edges of a segment. The border removes inconsistencies which enter a segment when it is translated, rotated, and scaled during the registration process.

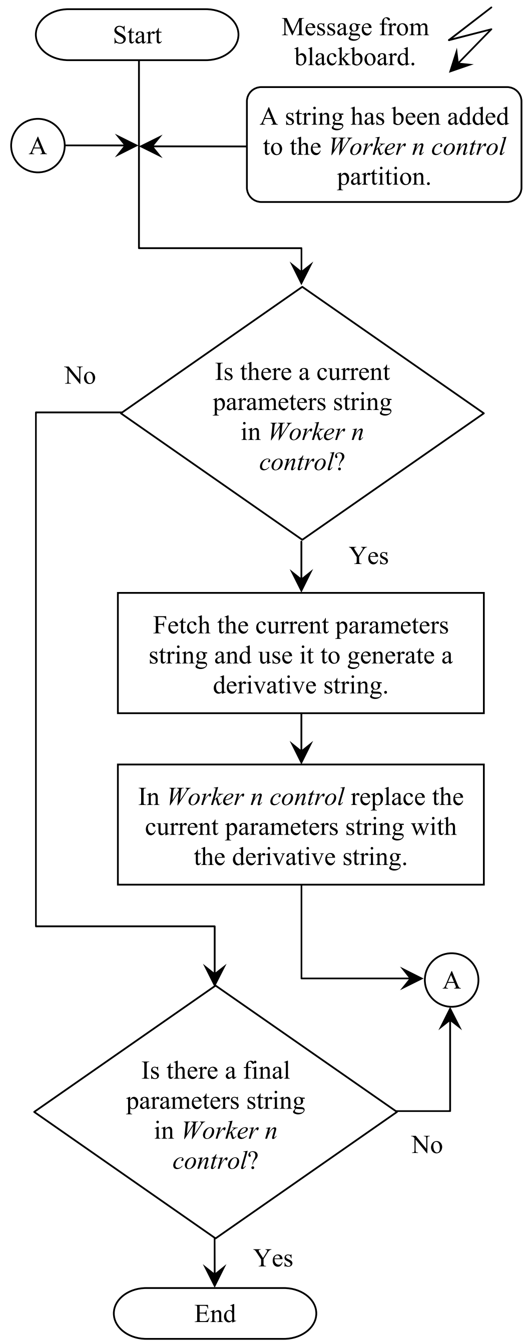
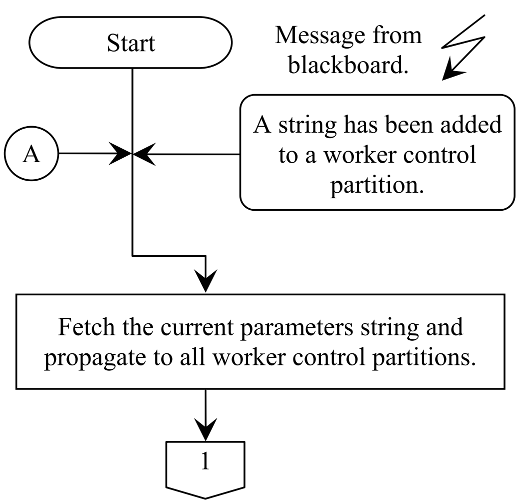


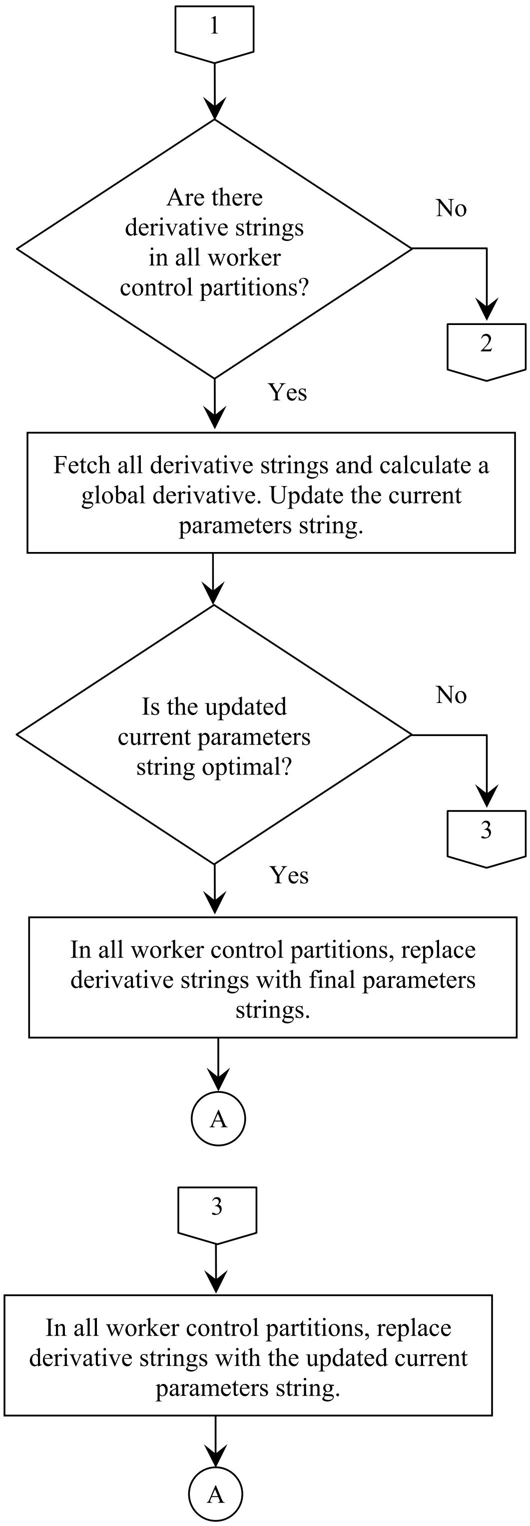
Fig 3. The Worker KS flow diagram illustrates the iterative retrieval of current transform parameters and the generation of derivative strings.

B. *The Worker n KS*

Connection to the blackboard and initialisation is the first job of the Worker n KS. The Worker n KS then waits for the current parameters string to appear in the *Worker n control* partition. As soon as the current parameters string appears fixed and moving segments are retrieved from the *Image container* partition. The region of interest string is also fetched from the *Worker n control* partition. These actions are part of a fire once mechanism that prevents the Worker n KS from further retrieval of fixed and moving segments during the optimisation process. Once the segments and region of interest have been retrieved a derivative string is generated. The string contains a local derivative which is calculated using parameters extracted from the current parameters string. The derivative string generated is used to replace the current parameters string in the *Worker n control* partition. This process is repeated every time a current parameters string appears. When the final parameters string appears the Worker n KS becomes inactive. Figure 3 illustrates the Worker n KS by means of a flow diagram.

In order for the Worker n KS to generate local derivatives of the similarity measure, for each intensity co-ordinates in the fixed segment corresponding moving segment co-ordinates are computed using parameters extracted from the current parameters string. If transformation of the fixed segment co-ordinates results in a corresponding location that fall inside of the moving segment a contribution to the local derivative is made, otherwise the intensity is considered invalid and the next intensity co-ordinates are evaluated. Contributions to the local derivative represent a summation of intensities from a gradient image, around the mapped co-ordinates. Using a recursive Gaussian gradient image filter, the gradient image is created from the moving segment. The gradient image represents a vector field in which every vector points in the direction of its nearest edge, an edge being a rapid increase or decrease in neighbouring intensities. Created once during initialisation of the Worker n KS, the gradient image is used for all iterations of the optimisation cycle.





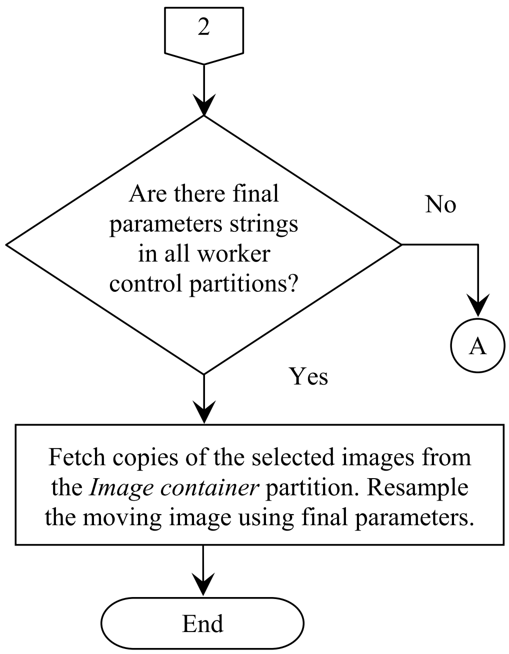


Fig 4. The Manager KS flow diagram illustrates the iterative generation of current transform parameters and their propagation to the worker control partitions.

C. *The Manager KS*

The Manager KS is the most complex of all framework components. The Manager KS starts by retrieving the current parameters string from the *Parameters* partition. The current parameters string is then propagated to all worker control partitions. Next, the Manager KS waits for local derivative strings to appear in all worker control partitions. On appearance of the local derivatives, the strings are accumulated and a global derivative calculated. Convergence tests that consider the magnitude of the global derivative and number of iterations performed are then conducted. On success of convergence testing a final parameters string is generated, otherwise the current parameters string is updated using the global derivative. The local derivative strings in all worker control partitions are then replaced with the newly updated or final transform parameter strings. On generation of the final parameters string, copies of the selected images stored in the *Image container* partition are retrieved. The moving image is then resampled using the final transform parameters. A flow diagram is given in Figure 4.

To update the current parameters string, the gradient-based optimisation scheme advances the current transform parameters in the direction of the global derivative. If the direction of the global derivative abruptly changes, it is assumed that an optimum has been encountered and the step length through the transform parameter search space is reduced by a half. After repeated iterations the step length is significantly reduced, thus restricting the selection of parameters to a small area of search space. Once step length becomes smaller than a predefined minimum the optimisation process is considered as having converged. This allows the precision of the final transform parameters to be specified.

IV. Experimental Results

Speed tests were conducted in order to demonstrate the performance increase of medical image registration in non-distributed and distributed processing environments. The spatial mapping of intensities during the registration process is achieved with a rigid 3D transform component. B-spline interpolation is used to evaluate moving intensities at non-discrete locations. To determine accuracy of alignment after the application of a transform, a normalised correlation similarity metric is provided. Optimisation of the computed similarity measure, using a search space defined by transform parameters, is achieved with a rigid 3D transform optimisation component.

In the non-distributed environment a sequential algorithm constructed from the same components, is hosted and run using a single processor. In the distributed environment, an algorithm is hosted using the registration framework described. For each experiment, the distributed testing represents an ideal case, i.e. one processor for the blackboard and one processor for each KS. Obtained from BrainWeb [12], testing was performed on three image pairs, each containing 181×217×180 voxels with a known translation and rotation. The selected images were divided by the Distributor KS into 1–14 segments and a 20-voxel wide border assigned. In all cases, times were combined and an average calculated.

A. *Sequential vs distributed normalised correlation*

Computed over all intensities in both images, normalised correlation calculates the intensity-wise cross-correlation of the images to be registered. Accurate alignment of images results in values near to one being generated. Misalignment, in contrast, produces values of less than one. The distributed similarity *S(F,M)* between fixed and moving images, as computed by the framework, is defined in (1)

$$S(F,M) = -1 \frac{\sum_{i=1}^{R}\left[\sum_{j=1}^{Q_i}\left(F(x_{ij}) - M(T(x_{ij}))\right)\right]}{\sqrt{\sum_{i=1}^{R}\left[\sum_{j=1}^{Q_i} F(x_{ij})^2 \sum_{j=1}^{Q_i} M(T(x_{ij}))^2\right]}} \qquad (1)$$

where F and M are fixed and moving segment intensity functions respectively. T is a spatial transform and x_{ij} is the j^{th} voxel of segment i from the fixed image. R is the number of segment an image is divided into and Q_i is the number of valid voxels between segments identified by i. The derivative $\partial S/\partial p$ of the normalised correlation similarity metric with respects to transform parameter p is computed using (2)

$$\frac{\partial S}{\partial p} = -\frac{\sum_{i=1}^{R}\left[\begin{array}{l}\sum_{j=1}^{Q_i}\left(F(x_{ij})\frac{\partial M(T(x_{ij},p))}{\partial p}\right)- \\ b\sum_{j=1}^{Q_i}\left(M(T(x_{ij},p))\frac{\partial M(T(x_{ij},p))}{\partial p}\right)\end{array}\right]}{a}$$

with

$$a = \sqrt{\sum_{i=1}^{R}\left[\sum_{j=1}^{Q_i}F^2(x_{ij})\sum_{j=1}^{Q_i}M^2(T(x_{ij},p))\right]} \tag{2}$$

$$b = \frac{\sum_{i=1}^{R}\left[\sum_{j=1}^{Q_i}F(x_{ij})M(T(x_{ij},p))\right]}{\sum_{i=1}^{R}\left[\sum_{j=1}^{Q_i}M^2(T(x_{ij},p))\right]}$$

where $M(T(x_{ij},p))$ represents a discrete input which has been interpolated using a B-spline interpolation scheme and $\partial M(T(x_{ij},p))/\partial p$ corresponds to a summation of intensities from a gradient image, around the mapped co-ordinates. Figure 5 shows results of speed testing with distributed normalised correlation as a similarity measure. As can be seen, the average execution time reduces from approximately 68 minutes to 9 minutes when ten Worker KSs are employed. This is a speedup factor of roughly 7 over the sequential algorithm. The distributed algorithm was observed to converge after the same number of iterations, with the same transform parameters as those computed by the sequential algorithm.

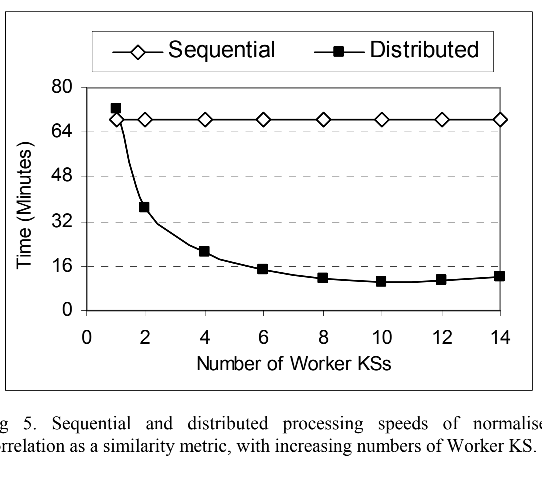


Fig 5. Sequential and distributed processing speeds of normalised correlation as a similarity metric, with increasing numbers of Worker KS.

V. Conclusions

Continued research has shown that concurrent similarity computation in medical image registration can be achieved and provides better speed performance than non-parallel implementations. In general, large speedups and high efficiency rates are historically difficult to achieve and only specialised hardware are capable of maintaining these rates when scaled. Unfortunately, the multi-processor architectures employed in the majority of research projects [6][7][8][9], restricts the usefulness of high performance algorithms to research labs. Also, due to the fine-grained parallelism employed, development of such distributed strategies is difficult. The framework described by this paper, in contrast, is based on a worker/manager model and provides an architecture that can reside on a network connected by the TCP/IP communication protocol. Although smaller speedup and scalability is achieved when compare with fine-grained parallelism, the coarse-grained approach employed has been shown to realise substantial performance increases when compared to a sequential implementation. Crucially, the modular nature of the architecture allows implementation of alternative similarity computation strategies, as specialised KSs, which can be added dynamically to the existing framework without changes.

References

[1] J.B.A. Maintz, and A. Viergever, “A survey of medical image registration,” *Medical Image Analysis*, no 2, pp 1–36, 1998.

[2] B. Zitova, and J. Flusser, “Image registration methods: A survey,” *Image and Vision Computing*, no 21, pp 977–1000, 2003.

[3] G.P. Penney, *et al.* “A comparison of similarity measures for use in 2D-3D medical image registration,” *IEEE Trans. Medical Imaging*, vol 17, no 4, pp 586–595, 1998.

[4] A. Roche, G. Malandain, N. Ayache, and S. Prima, “Towards a better comprehension of similarity measures used in medical image registration,” *Lecture Notes in Computer Science*, pp 555-566, 1999.

[5] S.K. Warfield, F. Jolesz, and R. Kikinis, “A high performance approach to the registration of medical imaging data,” *Parallel Computing*, vol 24, no 9–10, pp 1345–1368, 1998.

[6] G.E. Christensen, “MIMD vs SIMD parallel processing: A case study in 3D medical image registration,” *Parallel Computing*, vol 24, no 9–10, pp 1369–1383, 1998.

[7] T. Rohlfing, and C.R. Maurer, “Non-rigid image registration in shared-memory multi-processor environments with application to brains, breasts and bees,” *IEEE Trans. Information Technology in Biomedicine*, vol 7, no 1, pp 16–25, 2003.

[8] F. Ino, K. Ooyama, and K. Hagihara, “A data distributed parallel algorithm for non-rigid image registration,” *Parallel Computing*, vol 31, no 1, pp 19–43, 2005.

[9] M. Salomon, F. Heitz, G.R. Perrin, and J.P. Armspach, “A massively parallel approach to deformable matching of 3D medical images via stochastic differential equations,” *Parallel Computing*, vol 31, pp 45–71, 2005.

[10] L. Nolle, K. C. P. Wong and A. A. Hopgood, “DARBS: A distributed blackboard system,” *Research and Development in Intelligent Systems*, vol 18, pp 161-70, 2001.

[11] R.J. Tait, G. Schaefer, and A.A. Hopgood, “iDARBS - A distributed blackboard system for image processing,” *Proc. of 13th International Conference on Systems, Signals and Image Processing*, pp 431-434, 2006.

[12] Brainweb, http://www.bic.mni.mcgill.ca/brainweb.