Proceedings of the 28th IEEE
EMBS Annual International Conference
New York City, USA, Aug 30-Sept 3, 2006

Efficient 3-D Medical Image Registration Using a Distributed Blackboard Architecture

Roger J. Tait, Gerald Schaefer, Adrian A. Hopgood and Shao Ying Zhu

***Abstract*—A major drawback of 3-D medical image registration techniques is the performance bottleneck associated with re-sampling and similarity computation. Such bottlenecks limit registration applications in clinical situations where fast execution times are required and become particularly apparent in the case of registering 3-D data sets. In this paper a novel framework for high performance intensity-based volume registration is presented. Geometric alignment of both reference and sensed volume sets is achieved through a combination of scaling, translation, and rotation. Crucially, re-sampling and similarity computation is performed intelligently by a set of knowledge sources. The knowledge sources 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. Large-scale registrations with substantial speedups, when compared with a conventional implementation, have been demonstrated.**

I. Introduction

Transform optimisation, re-sampling, and similarity computation form the basic stages of an intensity-based registration process [1],[2]. During transform optimisation, translation, rotation, and scaling parameters which geometrically map points in the reference (fixed) volume to points in the sensed (moving) volume are estimated. Once estimated, voxel values which are mapped into non-integer co-ordinates are interpolated in the re-sampling stage. After re-sampling, a metric is used for similarity computation in which a degree of likeness between corresponding volumes is calculated. Optimisation of the similarity measure is the goal of the registration process and is achieved by seeking the best transform; the transform parameters therefore define the search space. Importantly, due to the iterative nature of registration algorithms, re-sampling [3] and computation of the similarity measure [4] represent a performance bottleneck which limits the speed of time critical applications.

To overcome the constraints associated with intensity-based registration, high performance computing has been employed by a number of researchers [5]–[7]. Clinically compatible speeds have been achieved by Warfield *et al.* [8] who introduced a parallel non-rigid algorithm based on the work-pile paradigm. In their research, a message passing interface and cluster of symmetric multiprocessors 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 [9] 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 their registration algorithm.

More recently, an approach based on multithreaded programming together with partitioning largely eliminated the need for explicit message passing between concurrent processes [10]. This scheme is reported to make implementation of high performance non-rigid registration a comparatively easy task when compared to other architectures, despite the need for specialised hardware. Using 64 CPUs (running at 400 MHz), registration of two 256×256×100 voxel datasets was achieved in approximately 1.5 minutes. A similar data distributed parallel algorithm is described by Ino *et al.* in [11]. 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 volume datasets as large as 1024×1024×590 voxels can be aligned in minutes rather than hours. Importantly, the limitations of memory space when processing large datasets is discussed in detail.

Building on previous work with a distributed blackboard architecture [12], the framework presented in this paper is designed to achieve high performance intensity-based volume registration. The innovative approach adopted supports multiple distributed knowledge sources (KSs) organised in a worker/manager model. By exploiting the blackboard's distributed nature, concurrent re-sampling and computation of the similarity measure, allows large-scale registration with substantial speedups to be realised. Crucially, the framework's modular architecture can be easily scaled to match performance requirements and allows for the addition of further, specialized, KSs as required. Comparison with a standard sequential algorithm confirms efficiency of our proposed approach.

R.J. Tait, G. Schaefer 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; gerald.schaefer@ntu.ac.uk; adrian.hopgood@ntu.ac.uk;).

S.Y. Zhu is with the Applied Computing Group, University of Derby, Derby, DE22 1GB, U.K., e-mail: s.y.zhu@derby.ac.uk.

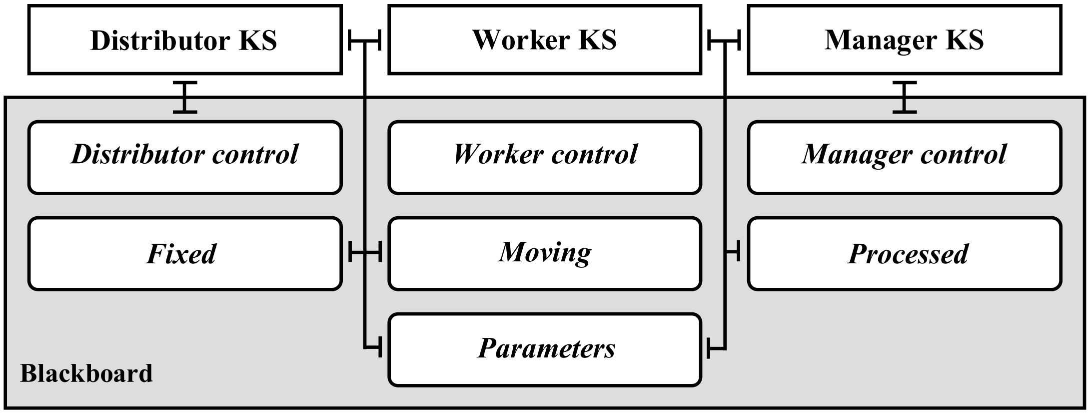

Fig. 1. Blackboard partitions used for logical storage and efficient retrieval of framework data. The interdependency between KSs and blackboard partitions is also shown. Due to the exhaustive search required, a drop in performance can be expected with a single-partition implementation. Similar inefficiency can be expected when a KS requests information through management and processing of excess partitions. The partitioning of data also aids design of the framework by introducing structure to an area of shared memory. This simplifies KS implementation as the number of partitions with which a KS works is kept to a minimum.

II. The Volume Registration Framework

The registration framework described consists of Distributor, Worker, and Manager KSs. Initialisation and volume selection are performed by the Distributor KS. On selection, fixed and moving volumes are split into segments which are placed on the blackboard. Local gradients are calculated by Worker KSs using the current transform and volume segments retrieved from the blackboard. When triggered, the Manager KS updates the current transform based on the local gradients while co-ordinating Worker KS activities. Calculation of local gradients and updating of the current transform is repeated until predefined thresholds are exceeded.

A. Partitioning of Framework Data

Partitioning of the blackboard is used to balance KS communication and processing workloads. Figure 1 illustrates the contents of the seven partitions the blackboard is divided into. The *Distributor control* partition supervises division of a volume into segments. The *Worker control* partition manages calculation of local gradients. A *Manager control* partition oversees updating of the current transform and supervision of Worker KSs. The *Parameters* partition hosts variables used by all KSs. Both *Fixed* and *Moving* partitions hold volume segments of their respective types.

B. Distributed Similarity Computation

To compute the similarity between fixed and moving segments, a distributed metric has been developed. During evaluation of the current transform, for each point in the fixed segment, a corresponding point in the moving segment is calculated. Interpolation is then used to compute an intensity value at the mapped position. By repeatedly summing resulting intensities for all valid voxels within a predefined region of interest, a local gradient is calculated by the Worker KS. Accumulation and summation of local gradients, performed by the Manager KS, allows for computation of a mean-squares-difference similarity measure. The distributed metric is an adaptation of a metric implemented as part of the ITK toolkit [13].

III. KS Implementation

The Distributed Algorithmic and Rule-based Blackboard System (DARBS) [14] is based on a client/server model where the server functions as the blackboard and KSs are implemented as client modules. Each knowledge source represents a structure in which rules and specialised algorithms can be embodied. Crucially, reading from and writing to the blackboard provides a mechanism for communication between KSs.

A. The Distributor KS

Placement of initial data on the blackboard and selection of volume datasets are two tasks performed by the Distributor KS. Data added to the blackboard includes the number of segments into which a volume is divided, a current transform, and regions of interest assigned to each segment. Importantly, regions of interest are added to their associated *Worker control* partition. Each region of interest is designed to remove non-voxel locations that enter at the edges of a segment due to translation and rotation during registration. The centres of mass are calculated for both fixed and moving volumes and the vector joining both centres is used as the initial transform and added to the *Parameters* partition. Division of volumes into segments and sending to their respective *Fixed* and *Moving* partitions is then performed.

B. The Worker KS

The first task of a Worker KS is initialisation and connection to the blackboard. Once connected, both fixed and moving segments with corresponding regions of interest are retrieved from the blackboard. The knowledge source then waits for the current transform to appear in its *control* partition. On appearance, the local gradient between the fixed and moving segments is then calculated, using the current transform and region of interest retrieved. The local gradient together with the number of valid voxels is then placed in the worker's *control* partition. Crucially, the local gradient and number of valid voxels is used in the computation of the similarity measure. This process (illustrated in Figure 2) is repeated until the current transform is replaced with a final transform by the Manager KS.

C. The Manager KS

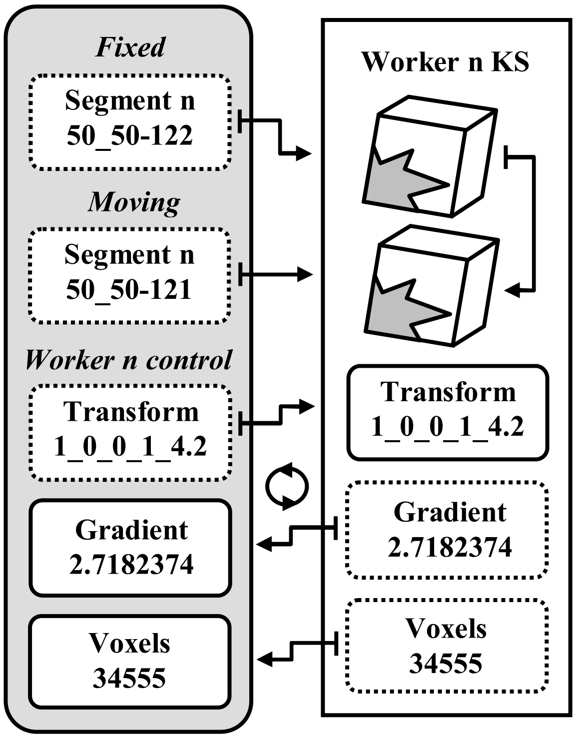

Fig. 2. The Worker KS engaged in iterative calculation of gradient and voxel data. To reduce duplicate and redundant data, once retrieved image segments and transform parameters are removed from the blackboard.

Once connected, the current transform placed in the *Parameters* partition is retrieved and used to initialise the Manager KS. The Manager KS then propagates the current transform to all *Worker control* partitions. It then waits for local gradients and valid voxel numbers to appear in all *Worker control* partitions. If local gradients or valid voxel numbers are missing no action is taken and the process is restarted. On retrieval of all local gradients and valid voxel numbers a similarity measure is calculated. The similarity measure (expressed as a double precision number) is used to update the current transform. As shown in Figure 3, convergence testing is then performed by the Manager KS which considers the updated transform length, magnitude of similarity measure, and number of iterations performed. If any of these parameters exceeds a predefined threshold the updated transform is replaced with a final transform; otherwise the updated transform is propagated to all *Worker control* partitions and hence another iteration of the registration process initiated.

D. Interaction of Framework Components

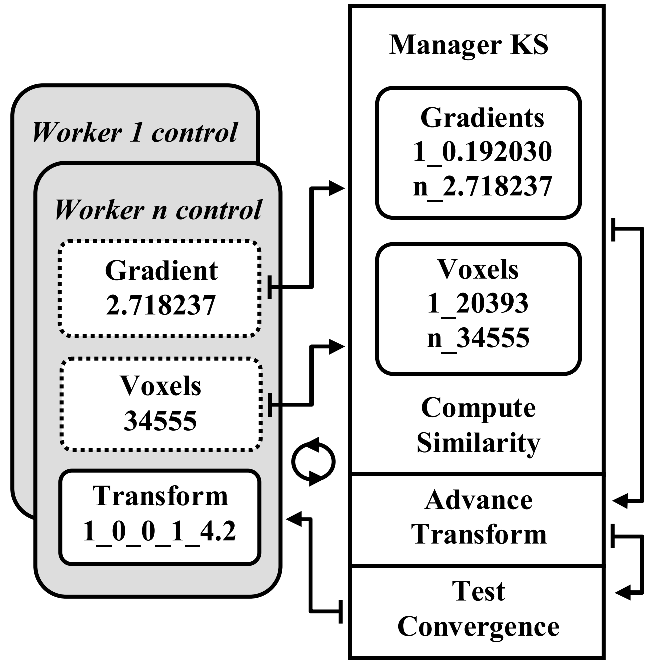

Fig. 3. The Manager KS employed in updating the transform parameters using gradient and voxel data retrieved from the blackboard. To prevent the Manager KS from continuously updating transform parameters, once retrieved gradient and voxel data are removed from the blackboard.

In our registration framework preconditions are attached to KS rule files and determine, in accordance with gradient and voxel data, when a KS can make its contribution to the optimisation process. This reactive behaviour removes the need for a dedicated control module and related overheads. Figure 4 shows the propagation of transforms to Worker KSs plus the flow of gradient and voxel data to the Manager KS.

IV. Experimental Results

To evaluate the increased performance of the registration framework quantitative evidence of its advantages over an alternative method currently in use was obtained. Components selected for testing of the framework included a versor-rigid-3D transform and b-spline interpolation. Versor-rigid-3D transform optimisation was employed to search iteratively for transform parameters that best satisfied the similarity metric. Using the ITK toolkit a performance benchmark for comparison, in the form of a sequential algorithm with the same components, was constructed. Obtained from BrainWeb [15], the data used for testing purposes consisted of 3 volume pairs, each containing 181×217×180 voxels with a known translation and rotation. Each volume pair was registered and the average over all runs calculated. All computers used for testing had 3GHz processors with 1GB of memory.

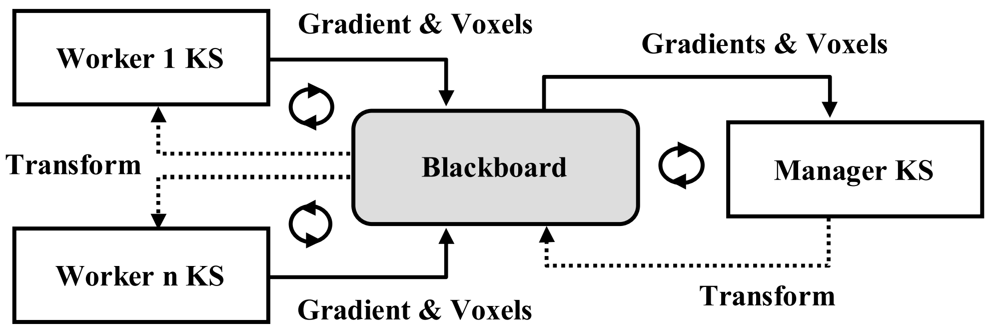

Fig. 4. Blackboard and KS interaction during the registration process. To ensure local gradients are calculated based on the current transform, the transform is removed from the blackboard when it is fetched by a Worker KS. Similarly, to ensure that updating of the current transform is based on gradients and voxel data from the same iteration, gradient and voxel data are removed from the blackboard.

The average execution times of both distributed and sequential registration approaches are presented in Figure 5. The results obtained prove a significant speedup – the average execution time is reduced from 27 to 5 minutes when 11 Worker KSs are employed – and hence confirm the applicability of our introduced framework for high performance registration of medical volume data.

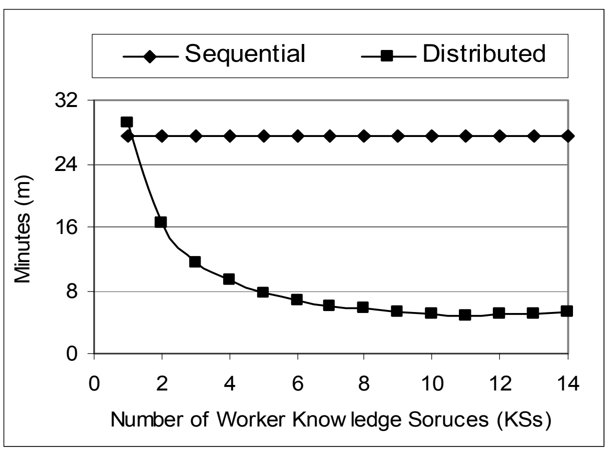

Fig. 5. Average sequential and distributed processing times based on three volume pairs with a known transform.

V. Conclusions

A framework for efficient registration of medical volume datasets was introduced. Based on a client/server blackboard architecture, gradient and similarity computation is effectively distributed to a number of worker knowledge sources. Experimental results on several brain volume datasets confirm that the concurrent volume re-sampling and calculation of the similarity measure allow significant speedups to be achieved. When compared to other schemes with dedicated hardware such as multiprocessors architectures our approach described here is cost-effective and can be easily expanded to meet increasing performance needs. Furthermore, the blackboard's modular nature allows implementation of alternative similarity computation strategies or of additional components 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. J. Grevera, and J. K. Udupa, "An objective comparison of 3D image interpolation methods," *IEEE Trans. Medical Imaging*, vol. 17, no. 4, pp. 642–652, 1998.
[4] 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.
[5] T. El-Ghazawi, P. Charlemwat, and J. Le-Moigne, "Wavelet-based image registration on parallel computers," *in Proc ACM/IEEE Conf. Supercomputing 97*, 1997, pp. 1–9.
[6] 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, no. 1, pp. 45–71, 2005.
[7] D. B. Russakoff, T. Rohlfing, and C. R. Maurer, "Fast intensity-based 2D-3D image registration of clinical data using light fields," in *Proc. 9th IEEE Inter Conf. Computer Vision*, 2003, pp. 416–423.
[8] 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.
[9] 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.
[10] T. Rohlfing, and C. R. Maurer, "Non-rigid image registration in shared-memory multiprocessor environments with application to brains, breasts and bees," *IEEE Trans. Information Technology in Biomedicine*, vol. 7, no. 1, pp. 16–25, 2003.
[11] 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.
[12] R. J. Tait, G. Schaefer, and A. A. Hopgood, "iDarbs - A distributed blackboard system for image processing," to appear in: *Proc. Int. Conf. Systems, Signals and Image Processing*, 2006.
[13] NLM Insight Segmentation and Registration Toolkit, http://www.itk.org.
[14] 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.
[15] Brainweb, http://www.bic.mni.mcgill.ca/brainweb.