

Intensity-Based Image Registration using Multiple Distributed Agents

Abstract

Registration is the process of geometrically aligning two images taken from different sensors, viewpoints or instances in time. A multiagent distributed blackboard system has been developed for intensity-based image registration for automated visual inspection. The images are divided into segments and allocated to individual agents on separate processors, allowing parallel computation of a similarity metric that measures the degree of likeness between reference and sensed images after the application of a transform. The need for a dedicated control module is removed by coordination of Distributor, Manager, and Worker agents through communication via the blackboard. Tests show that the system achieves large-scale registration with substantial speedups, provided the communication capacity of the blackboard is not saturated.

Key Words

Agents; DARBS; distributed blackboard system; knowledge-based system; parallel processing; image registration; similarity metric.

Introduction

In fields as diverse as manufacturing and medicine, there is an increasing need for automated visual inspection in the detection of defects or anomalies. The main motivating factors for the adoption of an automated approach include reliability, reproducibility, reduction of labour costs, and speed. In a manufacturing context, increased speed holds the potential of inspection rates matched to high-speed production.

A comprehensive overview of automated visual inspection for the detection of functional and cosmetic defects is provided by Newman and Jain [1]. The processing techniques can be grouped into referential comparison, non-referential modelling, and hybrid inspection [2]. In general, visual inspection is performed by moving samples in front of a camera. A high-resolution image is then captured and sent to a processing unit for analysis. Knowledge is extracted from fixed (i.e., reference) images and moving (i.e., captured) images in order that alignment and referential comparison can be made.

Many of the visual inspection methods reviewed in the literature employ image registration to align geometrically data taken from different sensors, viewpoints or instances in time [3]. During registration, fixed and moving images are aligned through a combination of translation, rotation, and scaling [4]. A universal registration technique is not possible due to the wide variety of noise and geometric deformations within captured data. Often these distortions are caused by the diverse methods of imaging available. Currently, registration is classified as either feature- or intensity-based, where both techniques have their own advantages and disadvantages.

Feature-based registration [5] is only as accurate as the initial selection of landmarks. In contrast, intensity-based registration methods [6] use all data within an image. Additional masking can be introduced to emphasise special features. The basic intensity approach consists of transform optimisation, image re-sampling and feature-matching stages. Feature matching is the most fundamental stage and is achieved through the use of a similarity metric [7], in which a degree of likeness between corresponding images is calculated. Both in practical and research terms, iterative computation of the similarity metric represents a considerable performance bottleneck that limits the speed of a visual inspection system. The flexibility of the inspection process is also limited by the inability to select between computational strategies for similarity based on their relative strengths, such as insensitivity to noise or possession of a large capture range [8].

The distributed multiagent framework presented in this paper achieves high-performance intensity-based image registration, which is a significant step in addressing the limitations of visual inspection. The innovative approach supports multiple distributed agents organised in a Worker/Manager model. Agent interaction and cooperation is achieved through the blackboard architecture, which has emerged from its 1970s origins as a modern, practical means of managing agent cooperation towards a common goal [9]. The original blackboard architecture was envisaged as a coordinated and distributed problem-solving environment that could be used to combine multiple processing techniques [10]. Advances in networking and agent-based technologies mean that this vision is now a reality. In the current work, rapid image registration is achieved, without recourse to expensive hardware, by sharing the computational task among separate agents that can run on any networked computer.

1 The Blackboard Architecture

A blackboard system is analogous to a team of experts who communicate their ideas via a physical blackboard, by adding or deleting items in response to the information that they find there. The experts were originally represented by specialist modules known as knowledge sources but, in a modern blackboard system, they are replaced by independent autonomous agents with specialized areas of knowledge.

Agents can communicate only by reading from or writing to the blackboard, a globally accessible working memory where the current state of understanding is represented. As each agent can be encoded in the most suitable form for its particular task, blackboard systems offer a mechanism for the collaborative use of different computational techniques such as rules, neural networks, genetic algorithms, and fuzzy logic. Each rule-based agent can use a suitable reasoning strategy for its particular task, e.g., backward- or forward-chaining, and can be thought of as a knowledge-based system in microcosm [9].

Since the emergence of the first blackboard architectures, most notably the Hearsay-II speech understanding system [11], a variety of frameworks have been employed in the inspection field. ARBS (Algorithmic and Rule-based Blackboard System) combined rules, algorithms, and neural networks for the interpretation of ultrasonic images [12]. ARBS has also been used to demonstrate coordinated problem solving through established techniques and through the novel Shifting Matrix Management model [13]. More recently, artificial neural networks embedded in a different rule-based blackboard system have been employed to identify erosion in steel bridge structures [14]. Both of

these examples are non-distributed architectures comprising three main components: the blackboard module, the agents, and a control or scheduler module.

DARBS (Distributed ARBS) [15, 16] is a distributed blackboard system based on a client/server model. The server functions as the blackboard while agents are implemented as client modules. The distributed nature of the implementation means that both blackboard and client modules run as separate processes and that no controller or scheduler is required. These independent processes may reside on a single processor or on any TCP/IP networked computers. Reading from and writing to the blackboard is implemented as standard functionality and provides a mechanism for communication between all agents. Storage of working data on the blackboard ensures equal access for all active agents.

2 Distributed Image Registration

Fig 1 shows that the image registration framework consists of Distributor, Manager, and N Worker agents. For an agent to be part of the framework it must first establish connection with the blackboard over the network. Framework initialisation and image selection are performed by the Distributor agent. The Distributor agent then splits fixed and moving images into segments before placing them on the blackboard. Worker agents take image segments from the blackboard and calculate local gradients using a global transform [17]. The Manager agent then updates the global transform based on local gradients, while coordinating Worker agent activities. Calculation of local gradients and updating of the global transform is repeated until predefined thresholds are exceeded. Finally, a resulting image is constructed from registered segments.

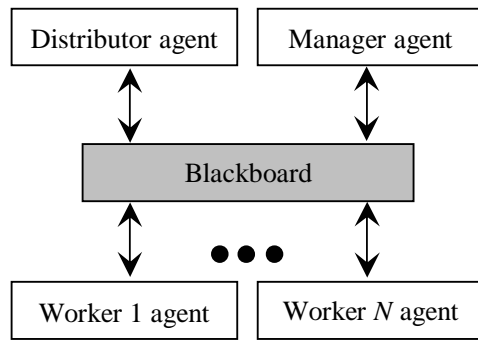


Fig 1 Worker/Manager model for the multiagent framework. The Distributor agent segments the image, Worker agents perform concurrent processing of the segments, and the Manager agent coordinates Worker agent activities.

2.1 Blackboard Partitions

On the blackboard, data partitioning is used to balance agent communication and processing workloads. Due to the exhaustive search required, a drop in performance can be expected with a single-partition implementation. Similar inefficiency can be expected when an agent requests information through management and processing of excess partitions. To combat these problems the chosen partition scheme allows interaction between agents in a logical and efficient manner. DARBS's unique ability to create, manipulate, and destroy partitions during run time overcomes the limitations of less dynamic blackboard implementations. The partitioning of data also aids design of the multiagent framework by introducing structure to an area of shared memory. This simplifies creation of agent rule files as the number of partitions with which an agent works is kept to a minimum.

As shown in Fig 2 the blackboard is initially divided into the following partitions:

- the *Distributor control* partition for controlling division of an image into segments;

- the *Worker control* partition for managing calculation of local gradients;
- the *Manager control* partition for managing updating of the global transform from local gradients and supervision of Worker agents;
- the *Parameters* partition for maintenance of variables used by all agents;
- both *Fixed* and *Moving* partitions for holding segments of their respective types.

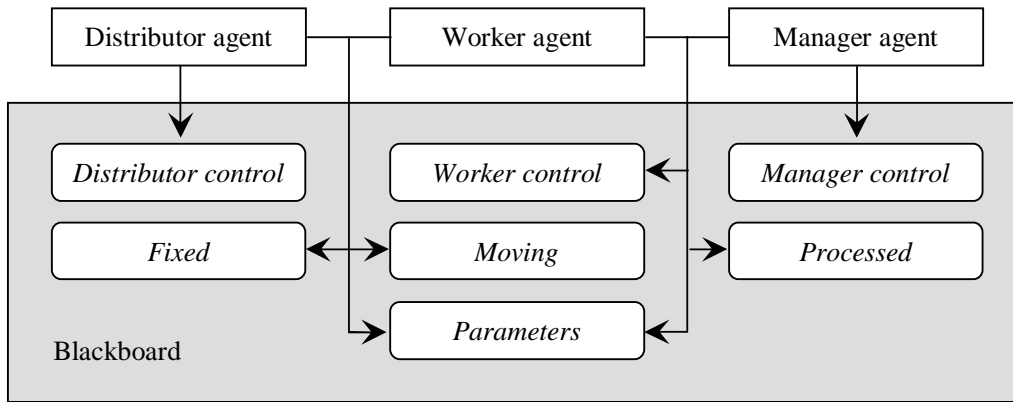


Fig 2 Blackboard partitions used for logical storage and efficient retrieval of data. Arrows indicate the specific partitions accessed by the individual agents.

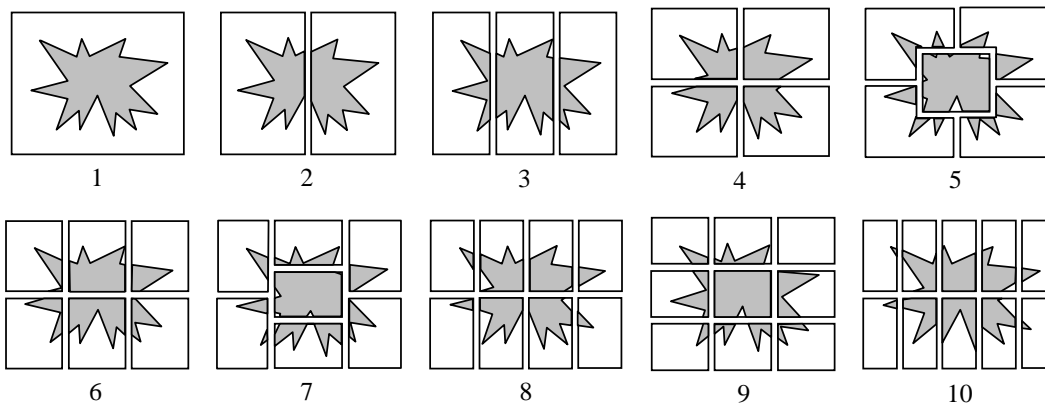


Fig 3 Image distribution scheme used to maximise detail appearing in each segment.

2.2 Data Transmission between Components

Image data are transmitted to and from the blackboard by the agents. Transmission data are divided into several parts. The first component corresponds to a segment identification number while the next values represent the segment's size. A minus symbol marks the end of attributes and the start of pixel data. The pixel format adopted is that of the unsigned char type. This significantly reduces the size of transmission data and associated overheads when compared with the more complex floating point type.

2.3 Distribution of Image Data

A distribution scheme was chosen whereby full resolution images are divided into a variable number of segments, each containing approximately the same number of pixels. Fig 3 shows how an image is split up into any number of segments between two and ten. This maximises the possibility of detail appearing in all segments and evenly distributes the workload between processors. The distribution scheme also benefits from the fact that no inter-processor communication is required. Duplication methods were not considered due to their transmission overheads.

2.4 Distributed Computation of Similarity

In a sequential registration process, a similarity metric is used to compute the degree of likeness between fixed and moving images after the application of a transform. To distribute similarity computation, two new metrics have been developed for the registration framework. They are adaptations of metrics implemented as part of the ITK toolkit [18]. During evaluation of a transform, for each point in the fixed segment, a corresponding point in the moving segment is calculated. The chosen interpolation scheme is then used to compute an intensity value at the mapped position. By repeating

the process within a predefined region of interest and summing resulting intensities for all valid pixels, a local gradient is calculated by the segment's assigned Worker agent n . Accumulation and summation of local gradients by the Manager agent allows for computation of either mean-squares (MS) or normalised-correlation (NC) similarity metrics between fixed (A) and moving (B) images. In the ITK implementations, MS and NS are defined as

$$MS = \frac{1}{N_p} \sum_{i=1}^{N_p} (A_i - B_i)^2 \quad (1)$$

$$NC = \frac{-\sum_{i=1}^{N_p} (A_i B_i)}{\sqrt{\sum_{i=1}^{N_p} A_i^2 \sum_{i=1}^{N_p} B_i^2}} \quad (2)$$

where A_i and B_i are the i^{th} pixels of the two images and N_p is the number of valid pixels considered. A pixel is thought valid if it maps to a position within the boundaries of the moving image. The new more efficient distributed version of these metrics, implemented as part of the registration framework, are

$$MS = \frac{\sum_{i=1}^S \left(\sum_{j=1}^{P_i} (A_j^i - B_j^i)^2 \right)}{\sum_{i=1}^S P_i} \quad (3)$$

$$NC = \frac{-\sum_{i=1}^S \left(\sum_{j=1}^{P_i} A_j^i B_j^i \right)}{\sqrt{\sum_{i=1}^S \left(\sum_{j=1}^{P_i} A_j^{i2} \sum_{j=1}^{P_i} B_j^{i2} \right)}} \quad (4)$$

where A_j^i and B_j^i are the j^{th} pixels of segment i from image A and B respectively, P_i is the number of valid pixels between segments identified by i , and S is the number of

segments into which the images are divided. The output from each metric is a similarity measure in the form of a double precision number.

3 Agent Implementation

Agents are provided with basic registration functionality through the embedding of shared library algorithms in rule files. To achieve this, simple image registration functionality was implemented in both Worker and Manager agents. An intensity-based algorithm suited to images of the same modality forms the basis of this functionality. The algorithm can be tailored, via the blackboard, to a specific problem with dynamically selectable components. The components consist of transform, interpolation and optimiser types. Both translation-only and centred-affine transform types are available to perform a spatial mapping between intensities in fixed and moving segments. In order to evaluate pixel values at non-grid positions, linear and b-spline interpolation schemes are provided. A gradient-descent optimiser is used to search iteratively for the transform that best satisfies the chosen metric.

3.1 The Distributor agent

The Distributor agent consists of the following four rule files:

- `Initialise_Distributor`
- `Select_Images`
- `Set_Transform`
- `Store_Segments`

Tasks performed by `Initialise_Distributor` include clearance of all data from the blackboard. The `Select_Images` rule causes a user interface consisting of a simple image viewer and Open File dialog box to appear. On selection of both fixed and moving

images, the user interface is automatically closed. Moments calculated from the selected images are used to estimate centres of mass. The vector that joins both centres is used as an initial transform and added to the *Parameters* partition. These actions form part of the *Set_Transform* rule. On firing of *Store_Segments*, the images are divided into segments and sent to their respective *Fixed* and *Moving* partitions. A region of interest is also generated for each segment and added to its associated *Worker control* partition. The region of interest is designed to create a border at the edges of a segment.

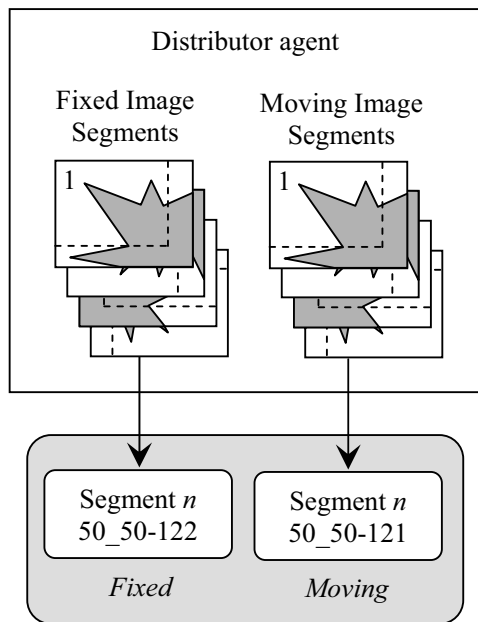


Fig 4 The Distributor agent divides an image into segments and places them in the fixed and moving blackboard partitions.

As illustrated in Fig 4, only edges that face neighbouring segments have a border. The border is intended to remove non-pixel values that enter at the edges of a segment, due to translation and rotation during registration. It also removes inconsistencies caused by convolution algorithms, such as the interpolator, that require a pixel's neighbourhood. Although the size of border is variable, the setting of a wide border will

cause a decrease in efficiency as additional redundant data will be accrued and processed.

3.2 *The Worker agent*

The Worker agent comprises of five rule files:

- Initialise_Worker
- Fetch_Segments
- Wait_Worker
- Perform_Optimisation
- Resample_Segment

Connection to the blackboard and initialisation of a Worker agent is performed by the Initialise_Worker rule. On firing of Fetch_Segments, both fixed and moving segments with a corresponding region of interest are retrieved from the blackboard. The Worker agent then enters a loop, by means of Wait_Worker, where it waits for a transform to appear in its *control* partition. This trigger mechanism ensures that all Worker agents commence processing at the same time.

Fig 5 shows addition and retrieval of data from the blackboard by a Worker agent. As soon as a transform appears in the Worker agent's *control* partition, it is removed. This stops the Worker agent from repeatedly firing the Perform_Optimisation rule. On firing of Perform_Optimisation, a local gradient between fixed and moving segments is calculated by the registration module using the fetched transform. Once calculated, the number of valid pixels and local gradient are placed in the Worker's *control* partition. The process is repeated each time an updated transform appears in the Worker agent's *control* partition. When a final transform appears, the Resample_Segment rule is fired,

causing translation and rotation of the moving segment using final transform parameters and return of the registered segment to the blackboard.

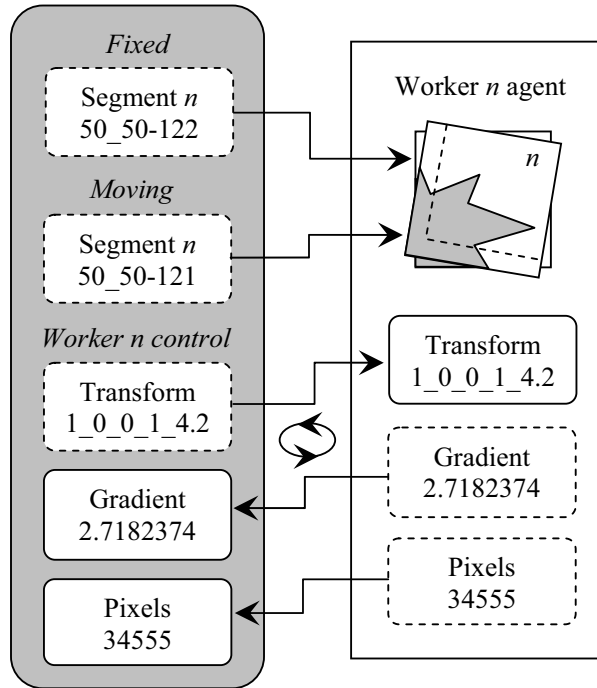


Fig 5 The Worker n agent iteratively collects transform parameters and calculates gradient and pixels data.

3.3 The Manager agent

The Manager agent consists of four rule files:

- Initialise_Manager
- Wait_Manager
- Advance_Transform
- Reconstruct_Image

The Manager agent is the most complex component of the framework. To initialise the Manager agent, on firing of Initialise_Manager, the initial transform placed in the

Parameters partition is retrieved. The initial transform is then propagated to all *Worker control* partitions. *Wait_Manager* causes the Manager agent to enter a loop, where it waits for the number of valid pixels and local gradients to appear in all *Worker control* partitions. No action is taken if pixel or gradient data are missing and the process is restarted.

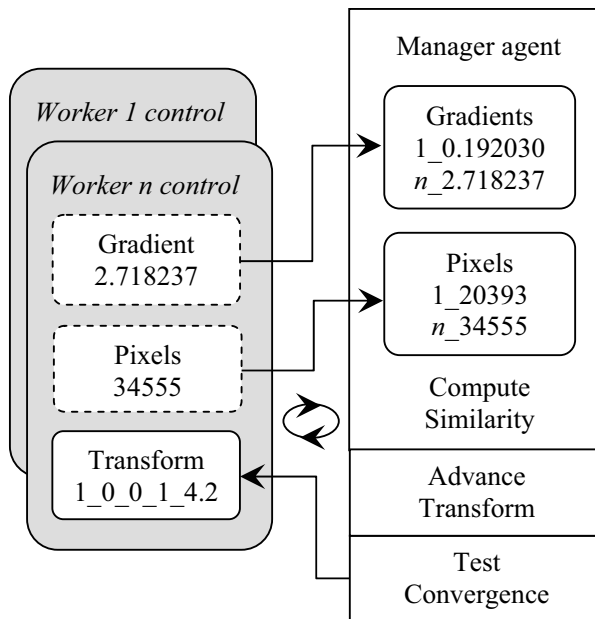


Fig 6 The Manager agent collects gradient and pixel data, which it uses to compute similarity, update transform parameters, and test if convergence of the registration process has been reached.

Addition and retrieval of data from the blackboard by the Manager agent is shown in Fig 6. A similarity measure is calculated using the total number of valid pixels and local gradients, on firing of *Advance_Transform*. The similarity measure is used to calculate an updated transform. A convergence test is then carried out that considers the updated transform's length, the magnitude of similarity measure and number of iterations

performed. In the event of these parameters exceeding a predefined threshold the updated transform is replaced with a final transform. Otherwise the updated transform is propagated to all *Worker control* partitions causing the gradient calculation and similarity evaluation processes to be repeated.

Reconstruct_Image is fired on appearance of the final transform. This rule causes registered segments to be retrieved from the blackboard. Each registered segment then has its borders removed before being inserted into a resulting image. The resulting image is automatically displayed by means of an image viewer.

3.4 *Interaction between agents*

As each agent adds to an iterative process, a control scheme was needed to coordinate contributions. Most blackboard implementations achieve coordination by the inclusion of a dedicated control module to activate specific agents. In the registration framework described here, preconditions are attached to agent rule files. These preconditions determine, in accordance with information on the blackboard, when an agent can make its contribution at any given time. This reactive behaviour removes the need for a dedicated control module and related overheads.

Fig 7 shows the iterative nature of both Manager and Worker agents during the registration process. The propagation of updated transforms to all Worker agents is shown, as well as the flow of gradients and pixels numbers to the Manager agent. As previously discussed, to ensure local gradients are calculated based on the same transform, the transform is removed from the blackboard when it is fetched by a Worker agent. To ensure that updating of the transform is based on local gradients and pixel numbers from the same iteration, local gradients and pixel numbers are removed from

the blackboard. A corrupt path through transform search space would occur if both Manager and Worker agents operated with parameters from different iterations.

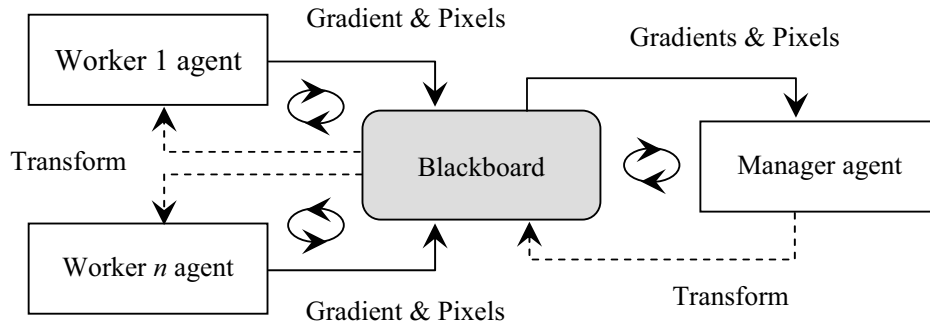


Fig 7 Flow of gradient data and updated transform parameters between agents.

Once updated by the Manager, transform parameters are propagated to all Worker agents.

4 Experimental Results

Registration into a common coordinate system requires iterative computation of a similarity metric before any referential comparison by an inspection system can be made. Accuracy of the registration process is also wholly dependent upon the selection of appropriate transform, interpolation and optimisation types. Components selected to perform testing of MS and NC metrics were therefore based on *a priori* knowledge. The following components were chosen:

- A centred-affine transform that allows rotation, scaling, shearing, and translation of image segments.
- b-spline interpolation in order to achieve greater accuracy than its linear counterpart.
- A regular step gradient descent optimiser because of its compatibility with other components.

In order to evaluate the increased performance of the registration framework, quantitative evidence of its advantages over an alternative method currently in use was required. A sequential algorithm, provided by the ITK toolkit, was updated with the same components and used as a benchmark for comparison.

4.1 Test Samples

Ultimately, the choice of resolution will determine the smallest size of detectable defect. Therefore large images, containing screen-printed bottle logos of approximately 1400×1800 pixels, were chosen as test samples. The fixed image represented a sample with an acceptable and verified quality of manufacture. In contrast, the moving images contained samples with a variety of defects. These included screen leak and missing print, both of which can be caused by incorrect ink viscosity, material contamination or wear. Subtraction before testing revealed that an unknown translation and rotation between fixed and moving images existed. In all cases, once selected, images were divided by the Distributor agent into segments and a 10-pixel wide border was assigned.

To simulate the referential comparison performed by an inspection system, additional image processing functionality was added to Worker agents. Now, on firing of the `Resample_Segment` rule, fixed and registered segments are thresholded using levels calculated by determining the between-class variance of each segment's intensity histogram [19]. A difference image is then created by subtracting the thresholded fixed segment from the registered segment. Finally, noise is removed from the difference image using morphological opening [20], this results in an opened segment which is returned to the blackboard in place of the registered segment.

An example of a fixed reference image, the corresponding moving image data, and a resulting registered segment after alignment by a Worker agent is shown in Fig 8. An area of missing print is clearly visible in the moving and registered segments. Translation and rotation caused by the alignment process has introduced non-pixel locations which are visible at the bottom (C) and right-hand sides (D) of the registered segment. These extraneous pixels are removed by the Manager agent when it constructs the resulting image.

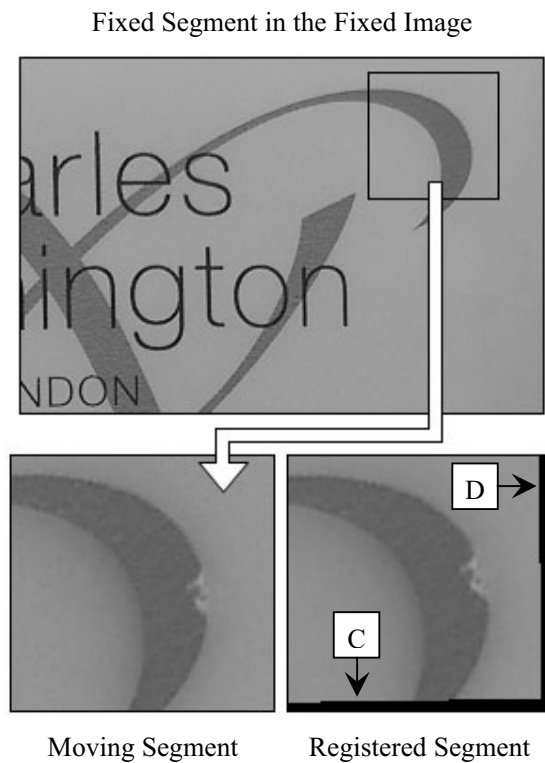


Fig 8 Fixed, moving and registered image data. An area of missing print is clearly visible in the moving and registered segments.

Fig 9 shows the difference segment created by subtracting the thresholded fixed image from the thresholded registered image. Phantoms (E and F) appear along the contours of the logo because the threshold level selection considers pixel intensities

only. This reliance on intensity makes segmentation particularly sensitive to changes in scene illumination and results in the extraction of features with slightly different boundary conditions. Even minor alignment differences after subtraction always appear as phantoms. The opened segment, also shown in Fig 9, illustrates how expansion and contraction of the difference segment results in the removal of such phantoms. Although slightly eroded, the segmented missing print (G) is clearly visible in the opened image, demonstrating that detection of minor defects has been achieved. To conserve small defects, the morphological structuring element used as an opening operator consisted of a single pixel. The opening operator can be used to eliminate both large and small phantoms through changes in size and shape of the structuring element.

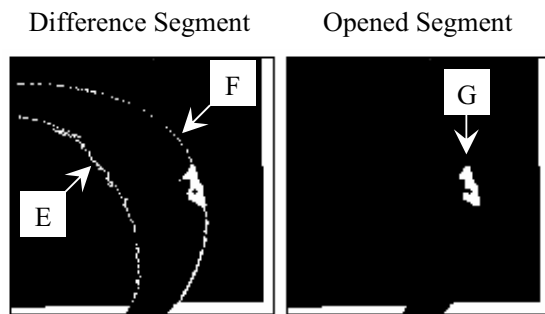


Fig 9 Referential comparison of the fixed and registered segments reveals a screen-printing defect.

To demonstrate the flexibility of the registration framework, printed circuit board [21] images were also tested. Fig 10 shows how samples containing artificially introduced defects can also be successfully registered and segmented by the framework. A spur (H) and an open circuit (I) have been detected. These are typical manufacturing defects that can be caused by dirt on the preprinted board or by air bubbles from electrolysis.

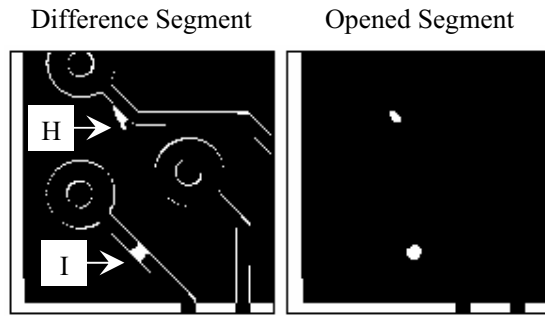


Fig 10 Referential comparison of the fixed and registered segments reveals printed circuit board defects.

4.2 Performance

Performance testing of the image registration framework was carried out in a computer laboratory with personal computers interconnected by an Ethernet 100Mbps switch. All computers in the network contained AMD Athlon 1.67GHz processors with 224 megabytes of random access memory and were running the Debian Sarge Linux operating system. During testing, the number of Worker agents was equal to the number of segments. Distribution of the framework also represented the ideal case, i.e. one processor for the blackboard and one processor for each agent. Each algorithm was applied to four images and the average processing time calculated. On convergence of the registration process, the number of iterations and final transform parameters were compared with the sequential implementation.

Fig 11 illustrates the sequential execution time and distributed speedup achieved during registration of bottle images with the MS similarity metric. It can be seen that the distribution of image data approximately reduces execution time from approximately 5 minutes to 110 seconds when seven Worker agents are employed. This maximum performance slowly diminishes as the number of processors increases.

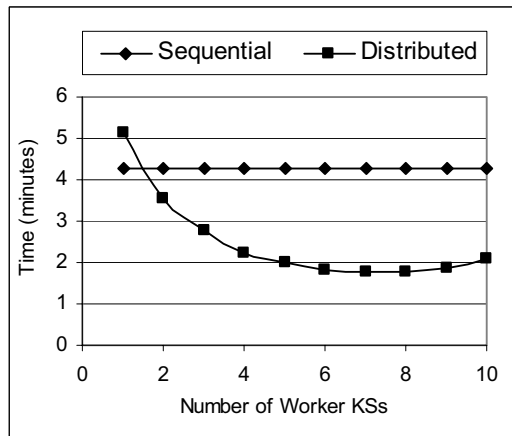


Fig 11 Sequential and distributed MS similarity metric performance with increasing numbers of Worker agents.

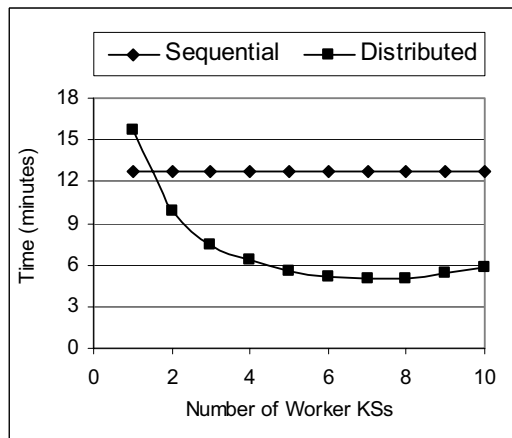


Fig 12 Sequential and distributed NC similarity metric performance with increasing numbers of Worker agents.

The sequential execution time and distributed speedup achieved during registration of bottle images with the NC similarity metric is given in Fig 12. It can be seen that the distribution of image data approximately reduces execution time from approximately 15 minutes to 220 seconds when seven Worker agents are employed. Again this maximum performance also slowly diminishes as the number of processors increases.

<i>Registration Parameters</i>	<i>Mean-squares (MS)</i>		<i>Normalised-correlation (NC)</i>	
	<i>Sequential</i>	<i>Distributed</i>	<i>Sequential</i>	<i>Distributed</i>
Iterations	19	19	31	31
Translation in X	13.806	13.806	13.681	13.681
Translation in Y	10.945	10.945	10.657	10.657
Rotation centre X	665.707	665.707	665.799	665.799
Rotation centre Y	896.997	896.997	896.983	896.983
Angle	9.95°	9.95°	9.898°	9.898°

Table 1 Registration parameters after convergence.

Table 1 provides the basic registration parameters for both MS and NC metrics after convergence. With each increase in the number of Worker agents, the framework converged after the correct number of iterations with transform parameters that matched those computed by the sequential algorithm. This correspondence of parameters indicates that the path through transform search space followed was the same for sequential and distributed implementations. Thus, the framework achieves increased performance when compared to an existing implementation. The reduced processing time of the MS metric, when compared to the NC implementation, is caused by its shorter path through transform search space. The shorter path and hence fewer iterations is reflected in the timescale of Figs 11 and 12.

5 Discussion

Computation of the similarity metrics is inherently parallel and well-suited to distributed implementation. The registration framework described in this paper shows that a distributed blackboard architecture, such as DARBS, is well-matched to such an implementation. Although successful in achieving performance increases, the increases diminish when image data are distributed between growing numbers of processors. Blackboard saturation occurs when the rate of requests by Worker and Manager agents to the blackboard becomes faster than the servicing of these requests. The saturation

effect is magnified by the shrinking size of image segments and increased communications to growing numbers of agents. One simple solution to this underlying problem is to increase the power of the blackboard processor. An alternative approach, which has been adopted by the registration framework described, is the reduction of control data stored on the blackboard.

Efficiency of the framework is reduced by an initial overloading of communications caused by Worker agents that try to obtain segments from the blackboard when first triggered. A second overload occurs when Worker agents have finished processing and try to return registered segments to their respective partitions. This synchronisation occurs when Worker agents operate in a first-come first-served fashion, particularly in situations when the number of segments is high. Creation of a schedule, prior to commencement of the Worker agents, represents a static load-balancing approach that could be adopted. Similarly, compression of transmission data should increase communication efficiency. The goal of any such approaches will be to better distribute communications and reduce the idle time of Worker agents.

6 Conclusions

High image resolutions, coupled with complex algorithms, have increased the demand for greater performance capabilities in the automated visual inspection field. Based on a Worker/Manager model and implemented using a distributed blackboard architecture, an innovative framework has been presented that achieves high-performance intensity-based image registration for use in referential comparison. Data partitioning and distribution, followed by computation of either MS or NC similarity metric, is achieved with specialised agents that work in parallel. The performed tests

clearly shows that parallel calculation of the similarity metric, which is seen as the major performance bottleneck associated with intensity-based registration, results in significant speedup compared with a non-distributed implementation. The approach described is cost-effective and can be easily expanded by the addition of agents and processors, unlike schemes with specialised hardware, such as shared-memory and multiprocessor environments. Other similarity computation strategies can be added as specialised agents, without changes to the framework.

For future work, it is intended that additional flexibility will be added to the framework in the form of metrics that are suited to images of differing modalities [22] and volumetric data [23]. Because some metrics need to be initialised with near optimal transform parameters while others have larger capture ranges, the selection of an appropriate metric for an inspection application will be dependent on the registration problem in hand.

References

1. T.S. Newman and A.K. Jain, A survey of automated visual inspection, *Computer Vision and Image Understanding* 61 (1995) 231-62.
2. E. Bayro-Corrochano, Review of automated visual inspection 1983 to 1993 part II: approaches to intelligent systems. *Proceedings of SPIE Intelligent Robots and Computer Vision* (1993) 159-72.

3. B. Zitova and J. Flusser, Image registration methods: a survey, *Image and Vision Computing* 21 (2003) 977-1000.
4. L.G. Brown, A survey of image registration techniques, *ACM Computing Surveys* (1992) 325-376.
5. B. Temkin, S. Vaidyanath and E. Acosta, A high accuracy, landmark-based, sub-pixel level image registration method, *International Congress Series* 1281 (2005) 254-259.
6. K. Jeongtae and J.A. Fessler, Intensity-based Image Registration using Robust Correlation Coefficients, *IEEE Transactions on Medical Imaging* 23 (2004) 1430-1444.
7. G.P. Penney, J. Weese, J.A. Little, P. Desmedt, D.L.G. Hill and D.J.A. Hawkes, Comparison of similarity measures for use in 2D-3D medical image registration, *IEEE Transactions on Medical Imaging* 17 (1998) 586-95.
8. J. Zhang and A. Rangarajan, Affine image registration using a new information metric, *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'04)* (2004) 848-855.
9. A.A. Hopgood, The state of artificial intelligence, *Advances in Computers* 65 (2005) 1-75.
10. H.P. Nii, Blackboard systems: the blackboard model of problem solving and the evolution of blackboard architectures, *AI Magazine* 7 (1986) 38-53.

11. L.D. Erman, F. Hayes-Roth, V.R. Lesser and D.R. Reddy, The Hearsay-II speech understanding system: integrating knowledge to resolve uncertainty, *ACM Computing Surveys* 12 (1980) 213-253.
12. A.A. Hopgood, N. Woodcock, N.J. Hallam and P.D. Picton, Interpreting ultrasonic images using rules, algorithms and neural networks, *European Journal of Non-Destructive Testing* 2 (1993) 135-149.
13. G. Li, A.A. Hopgood and M.J. Weller, Shifting matrix management: a model for multi-agent cooperation, *Engineering Applications of Artificial Intelligence*, 16 (2003) 191-201.
14. S.V. Barai and P.C. Pandey, Integration of damage assessment paradigms of steel bridges on a blackboard architecture, *Expert Systems with Applications*, 19 (2000) 193-207.
15. L. Nolle, K.C.P. Wong and A.A. Hopgood, DARBS: a distributed blackboard system, *Research and Development in Intelligent Systems XVIII*, M. Bramer, F. Coenen and A. Preece (eds.) (2001) 161-70.
16. K.W. Choy, A.A. Hopgood, L. Nolle and B.C. O'Neill, Implementation of a tileworld testbed on a distributed blackboard system, *18th European Simulation Multiconference* (2004) 129-135.
17. R.J. Tait, G. Schaefer and A.A. Hopgood, Towards High Performance Image Registration Using Intelligent Agents, *13th International Conference on Systems, Signals and Image Processing (IWSSIP)*, Budapest, Hungary (2006).

18. NLM. Insight segmentation and registration toolkit, <http://www.itk.org> (2004).
19. N. Otsu, A threshold selection method from gray-level histograms, *IEEE Transactions on Systems Man and Cybernetics* (1979) 62-66.
20. S. Umbaugh, *Computer Vision and Image Processing*, Prentice Hall USA (1998).
21. M. Moganti, F. Ercal, C. Dagli and S. Tsunekawa, Automatic PCB inspection algorithms: a survey, *Computer Vision and Image Understanding* 63 (1996) 287-313.
22. C. Nikou, F. Heitz and J. Armspach, Robust voxel similarity metric for the registration of dissimilar single and multimodal images, *Pattern Recognition* 32 (1999) 1351-1368.
23. W. Wells, P. Viola, H. Atsumi, S. Nakajima and R. Kikinis, Multi-modal volume registration by maximization of mutual information, *Medical Image Analysis* 1 (1996) 35-51.