

DEFECT DETECTION USING A DISTRIBUTED BLACKBOARD ARCHITECTURE

Roger J. Tait, Gerald Schaefer, Adrian A. Hopgood and Lars Nolle
School of Computing and Informatics
Nottingham Trent University

Burton Street, Nottingham, UK, NG1 4BU

e-mail: roger.tait@students.ntu.ac.uk, {gerald.schaefer,adrian.hopgood,lars.nolle}@ntu.ac.uk

KEYWORDS

Defect detection, blackboard system, parallel image processing, image registration.

ABSTRACT

The framework of an automated vision system for the monitoring of quality control is presented. Inspection which is capable of detecting various forms of defects is achieved by combining distributed artificial intelligence and image processing. The blackboard architecture DARBS (Nolle and Wong 2001) manages the processing of image data via an area of shared memory where the current understanding of the problem evolves. Registration into a common co-ordinate system and segmentation of reference and sensed images is performed by intelligent agents which communicate with each other by means of the blackboard. Pixel-level fusion is performed on registered images in order to exploit complementary and redundant data, allowing identification of suspected defects. The difficulties of landmark extraction common to feature-based registration techniques have been replaced by an intensity-based algorithm. Addition or removal of specialised agents is simplified by the blackboard's modular nature.

1. INTRODUCTION

In manufacturing industries there is an increasing need for automated detection and characterisation of defects. The motivating factors for the adoption of an automated inspection approach include the reduction of expensive labour costs, reproducibility, and the matching of high-speed inspection with high-speed production. Identification of functional and cosmetic defects in finished products has been achieved using a number of techniques, a general overview of automated visual inspection is provided by (Newman and Jain 1995). The processing techniques described can be grouped into referential comparison (Lee 1978), non-referential modelling (Wen and Tao 1999) and hybrid inspection (Bayro-Corrochano 1993).

A variety of approaches in the printed-circuit-board (PCB) production environment are described by (Moganti et al. 1996). One such technique (Gokturk et al. 1999) uses the differentiating characteristics of a design, stored as a library, to compare and detect defects in a captured image. The detection of defects in

the production of moulded plastic products which employ a Fourier descriptor and differential gradient operator to classify imperfections as being either shape or surface anomalies is described by (Petkovic et al. 2002). Despite the advantages of speed, consistency and accuracy of an automated system over manual inspection, often the objective of these approaches is to ensure electrical connectivity, check form and classify quality using an inspection technique that consists of a rigidly defined set of steps.

The framework presented in this paper will monitor the quality of a sample and be capable of providing information for the removal of a defective product further along a production line. The blackboard architecture supports multiple agents each of which performs an independent task. By exploiting the opportunistic nature of the blackboard architecture the framework is freed from a predefined processing path and allows for the possibility of integrating different inspection techniques.

2. THE BLACKBOARD ARCHITECTURE

DARBS is a distributed blackboard system based on a client/server model. The server functions as the blackboard and client modules as knowledge sources (KSs). During execution the blackboard is used to host evolution of a solution to a problem. A KS in contrast is a structure in which rules and algorithms can be embodied. Reading from and writing to the blackboard is implemented as standard KS functionality. When updates to the solution of a problem are made, a broadcasting of messages is used to inform all relevant KSs. A KS's behaviour guides it on a course of action in response to broadcasted messages.

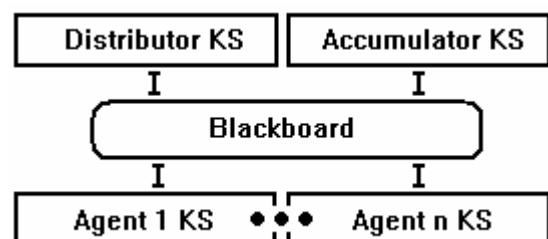


Figure 1: Blackboard components

Figure 1 shows that the initial implementation of the framework presented in this paper consists of a

Distributor KS, several Agent KSs and an Accumulator KS. The Distributor KS splits an image into segments before placing these on the blackboard. The Agent KSs takes image segments from the blackboard, registers them and processes them according to their implemented behaviour. A resulting output image is constructed from processed image segments by the Accumulator KS.

As the current state of a problem is stored on the blackboard, partitioning is used to balance communication and processing workloads. The partitioning of data aids the design of the framework by introducing structure to the blackboard. 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. To combat these problems, the chosen partition implementation allows interaction between KSs in a logical and efficient manner.

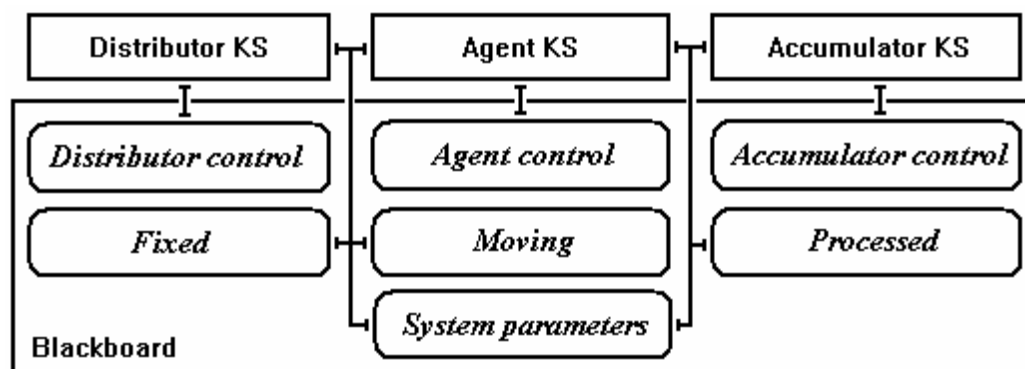


Figure 2: Blackboard partitions

Figure 2 shows that the blackboard is divided into the following partitions:

- A *Distributor control* partition controls division of an image into segments.
- The *Agent control* partition is used to manage processing of image segments.
- Reconstruction of the resulting image is accomplished within an *Accumulator control* partition.
- Variables used by all KSs are maintained by a *System parameters* partition.
- *Fixed*, *moving* and *processed* partitions hold image segments of their respective types.

Segment data is transmitted to and from the blackboard by the KSs. Transmission data is formatted into several parts. The first component corresponds to an image identification number, the next two values represent the image size. A minus symbol marks the end of attributes and the start of pixel data.

3. KS IMPLEMENTATION

By changing the structure of a KS's rules and through provision of additional functionality, the behaviour of a KS can be altered. The ITK toolkit (NLM 2004) was used to provide an Agent KS with registration and segmentation functionality through the embedding in rule files of shared library algorithms. Three simple image processing modules which interface ITK with the framework have been created; they encapsulate registration, segmentation and fusion functionality.

A multi-resolution registration algorithm which is capable of aligning images produced by a range of modalities forms the basis of the registration module. The algorithm can be tailored to a specific problem with dynamically selectable transform, interpolation, metric and optimiser components. Global and automatic thresholding functionality to separate an object from its background is provided through the segmentation module. Morphological operators for noise removal are also part of the segmentation module.

The fusion module performs an exclusive-or operation of image data. This XOR operation is used by Agent KSs to identify immediate differences between images and hence potential defects.

3.1 The Distributor KS

Several tasks are performed by the Distributor KS. The first task involves the initialisation of data in the *System parameters* partition. The data includes the size of borders between adjacent segments and the number of segments an image is to be divided into. Registration parameters of transform, interpolation, metric and optimiser are also stored in the *System parameters* partition.

The second task involves selection of reference (fixed) and sensed (moving) images. In order to make sure all segments will be of the same size, both fixed and moving images are resized. If required, pixels from the bottom and right hand sides of an image are removed. Resizing of the images simplifies the production of a

resulting image. Both images are then divided into segments by the Distributor KS and sent to their respective partitions on the blackboard. Before transmission of data, an identification number is assigned to each segment.

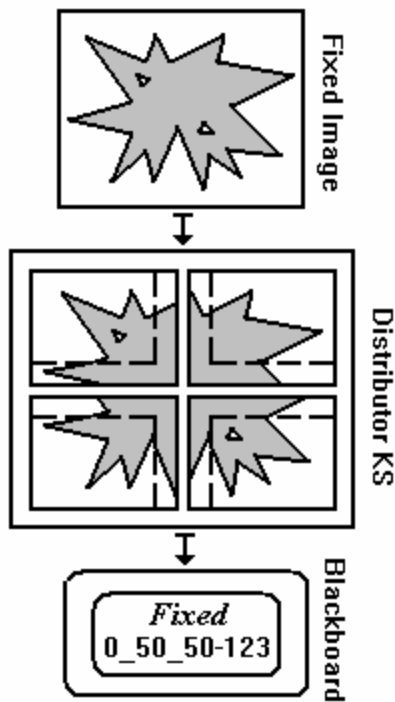


Figure 3: Distributor KS

Figure 3 shows the dividing of an image by the Distributor KS. Since the number of image segments is variable a balanced communication workload can be maintained. An increase in computational overhead however will occur when a large number of segments are allocated. Similarly, the objective of parallel processing will be defeated when the number of segments is small. Figure 3 also shows that only edges which face neighbouring segments have a border. The border is designed to remove non-pixel values which enter at the edges of a segment due to translation during registration. The border also removes the inconsistencies caused by convolution algorithms which require each pixel's neighbourhood.

3.2 The Agent KS

Once initialised an Agent KS enters a loop. A trigger mechanism ensures that all active Agent KSs commence processing at approximately the same time. Triggered Agent KSs then process segments in the following way:

- The current image segment waiting to be processed identification variable (CWP) is retrieved from the blackboard.
- Fixed and moving segments with identification numbers matching CWP are retrieved from the blackboard.
- CWP variable is incremented and returned to the blackboard.

- Fixed and moving segments together with registration parameters fetched from the blackboard are passed to the registration module.
- The process repeats itself until no segments are left for processing.

The registration parameters consist of transform, interpolation, metric and optimiser types. Both translation only and affine transform types are available to perform a spatial mapping between points in the fixed and moving segments. In order to evaluate pixel values at non-grid positions bi-linear and nearest neighbour interpolation algorithms have been implemented. Either normalised cross correlation or mutual information metrics can be used to measure the match between segments after they have been transformed. A gradient decent optimiser is used to iteratively search for the transform which best satisfies the chosen metric. According to modality, segmentation and fusion occurs once a segment has been registered.

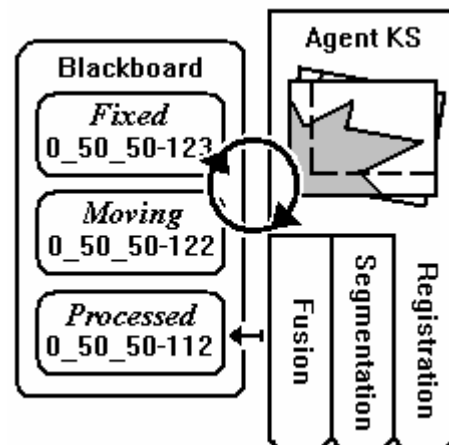


Figure 4: Agent KS

Figure 4 shows the registration of fixed and moving segments by an Agent KS. Once registered a segment is sent to the *Processed* partition on the blackboard. Processing segments in a first-come first-serve fashion allows a fixed number of Agent KSs to process a fixed number of segments. Alternatively a variable number of Agent KSs. This flexibility of processing means when an error is encountered a graceful degradation of the framework can occur. Importantly, memory usage of the blackboard is kept to a minimum by removing segments once they have been retrieved from the blackboard.

3.3 The Accumulator KS

The Accumulator KS is the simplest of all framework components. Once initialised the Accumulator KS stores in the *Accumulator control* partition a segments gathered variable (SG). The Accumulator KS then

enters a loop and process segments in the following way:

- Segments with identification numbers corresponding to SG are fetched from the blackboard.
- SG is incremented each time a segment is successfully retrieved.
- Retrieved segments are added to a list which is locally maintained by the Accumulator KS.
- The process repeats itself until all segments have been processed.

The total number of segments and border size variables are retrieved from the blackboard, as soon as all segments have been gathered. Each segment is then taken from the locally maintained list. Its borders are removed and the segment is inserted into the resulting image. Finally, the constructed resulting output image is automatically displayed by means of an image viewer.

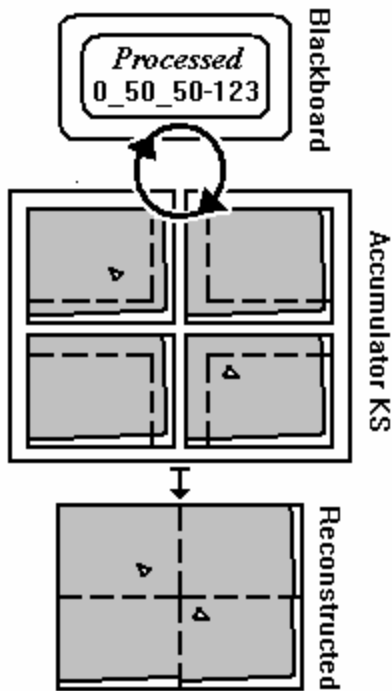


Figure 5: Accumulator KS

Figure 5 shows how the Accumulator KS assembles a resulting image by removing segments from the *Processed* partition. The resulting image contains regions of localised rather than global registration. This is because the registration algorithm applies a transformation to each segment. A confidence level based on the magnitude of difference present in a segment can therefore be assigned. To maintain consistency the locally maintained segment list is cleared of all data whenever the Accumulator KS is started and after construction of a resulting image. Again, graceful degradation of the framework in the event of an error is possible by gathering segments in this way.

4. EXPERIMENTAL RESULTS

Image subtraction behaviour was added to the Agent KSs using the image processing modules in order to evaluate the initial detection performance of the framework and registration precision. PCB images of approximately 1000x500 pixels were chosen as test samples. The fixed image represents a sample with an acceptable and verified quality of manufacture. In contrast, the moving image is a sample containing a variety of defects. These include a spur and an open circuit both of which can be caused by dirt on a blank board or by air bubbles from electrolysis. Once selected, both images were divided by the Distributor KS into 4 segments and a 10-pixel wide border was assigned.

4.1 Registration

Spatial alignment into a common co-ordinate system of both fixed and moving segments was performed before the referential comparison could be made. The following components were chosen to perform registration:

- Affine transformation which allows for translation, rotation, and scaling of segments.
- Linear interpolation in order to allow pixel intensities to vary continuously between grid positions.
- Cross correlation metric in order to perform a pixel-wise association between the two segments.
- A regular step gradient descent optimiser because of its compatibility with the other components.

4.2 Segmentation

Once registered, threshold levels were automatically determined for fixed and registered segments. In order to segment the conductor from the insulation material, both segments were globally thresholded.

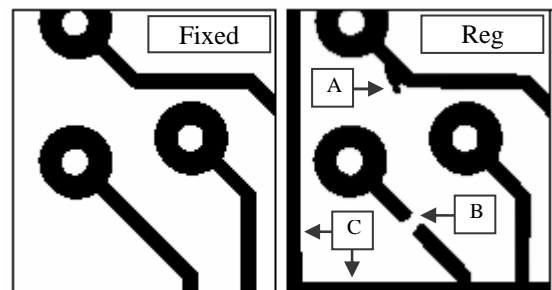


Figure 6: Thresholded segments

Figure 6 shows how both spur (A) and open circuit (B) defects have been segmented and are clearly visible in the registered segment. Additionally, translation and rotation caused by the registration process have introduced non-pixel locations which are visible at the bottom and left-hand sides (C) of the segment. All

extraneous pixels at the segment's borders are removed by the Accumulator KS.

4.3 Fusion

For the purposes of testing, subtraction was implemented as an exclusive-or operation between fixed and registered segments. Regardless of the sub-pixel accuracy achieved by the registration algorithm, differences appeared after subtraction as phantoms.

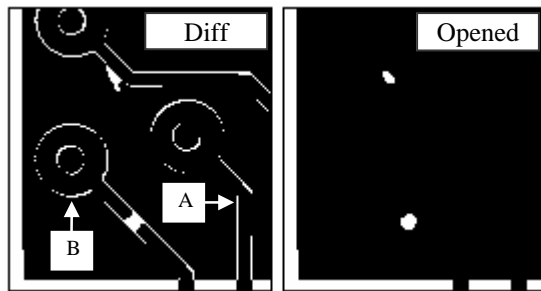


Figure 7: Difference and opened segments

Figure 7 shows that these phantoms appear along the contours of the conductive material (A,B). As the segments are of a relatively high resolution, phantoms which appear are generally smaller in size than potential defects. In order to achieve better segmentation, morphological opening was applied to the difference segment. Expansion and contraction of the segment caused by the opening operator resulted in the removal of all phantoms. To conserve small defects, the morphological structuring element 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.

5. CONCLUSIONS

The initial implementation of an automated visual inspection framework has been described. As the choice of resolution determines the smallest size of detectable defect, high resolutions have been used to prevent loss of detail during the processing of a difference image. The difference image is produced by subtracting fixed and moving images after registration into a common co-ordinate system. Preliminary testing of the intensity based registration algorithm showed that to obtain reasonable processing speeds, images needed to be small or down sampled in size. Large high resolution images were found to take more than a minute to register. In contrast, smaller images required only a few seconds. To address the problems of resolution and speed the blackboard architecture provides a distributed approach. The blackboard allows for an image to be divided into a number of smaller segments which are distributed between multiple agents. Segments are processed concurrently so that high sensitivity and increased speeds can be maintained. It is intended that accurate detection of

smaller defects will be achieved through experimentation with greater resolutions.

REFERENCES

- Bayro-Corrochano, E. 1993. "Review of automated visual inspection 1983 to 1993 Part II: Approaches to intelligent systems." *SPIE: Intelligent Robots and Computer Vision*, 159-172.
- Gokturk, S. Akarun, L. and Bozma, I. 1999. "Automated Inspection of Printed Circuit Boards Using a Novel Approach". *IEEE Workshop on Nonlinear Signal and Image Processing*.
- Lee, D.T. 1978. "A Computerized Automatic Inspection System for Complex Printed Thick Film Patterns." *SPIE: Applications of Electronic Imaging Systems*, 143, 172-177.
- Moganti, M. Ercal, F. Dagli, C.H. and Tsunekawa, S. 1996. "Automatic PCB Inspection Algorithms: A Survey." *Computer Vision and Image Understanding*, (63), 287-313.
- Newman, T.S. and Jain, A.K. 1995. "A Survey of Automated Visual Inspection." *Computer Vision and Image Understanding*. 61(2), 231-262.
- NLM Insight Segmentation and Registration Toolkit, <http://www.itk.org>, 2004.
- Nolle, L. Wong, K.C.P. and Hopgood, A.A. 2001. "DARBS: A Distributed Blackboard System." *Research and Development in Intelligent Systems*, XVIII, 161-170.
- Petkovic, T. Krapac, J. Loncaric, S. and Sercer, M. 2002. "Automated visual inspection of plastic products." *Proceedings of ERK 2002*, 283-286.
- Wen, Z. and Tao, Y. 1999. "Building a Rule-based Machine Vision System for Defect Inspection on Apple Sorting and Packing Lines." *Expert Systems with Applications*, (16), 307-313.