# Towards intelligent boundary extraction for on-line industrial inspection

**Q. M. Wu and M. G. Rodd**
*Department of Electrical and Electronic Engineering, University of Wales, Swansea, Swansea SA2 8PP, UK*
*(Received June 1989)*

## ABSTRACT

Recently, increasing demands in industry and many other application areas for inexpensive image-processing equipment have caused an upsurge of interest in real-time computer vision—particularly in terms of applications in on-line process control or in direct manufacturing control. The development of new computer vision algorithms and flexible techniques which also demonstrate a computational complexity that will make real-time applications economically feasible in the near future, is particularly attractive. This paper presents a computer inspection system which is applicable to real industrial applications. The major contribution of the paper is the development of an integrated, knowledge-driven and intelligent boundary extractor, which can extract closed boundaries of touching objects directly from grey-level images in real-world, industrial environments, and various fast techniques for verification of object structures at on-line speeds.

## INTRODUCTION

Computer vision can, potentially, be used to play a very important role in many closed-loop control situations—particularly in terms of applications in on-line process control or in direct manufacturing control. The introduction of an external sensing mechanism within a robot control loop can make robots more intelligent and flexible. It allows a robot to interact with its environment in a flexible manner. A robot that can 'see' and 'feel' is easier to train in the performance of complex tasks while, at the same time, requiring less-stringent control mechanisms than preprogrammed machines. A sensory, trainable system is also adaptable to a much larger variety of tasks, thus achieving a degree of universality that ultimately translates into low production and maintenance costs. While proximity, touch, and force sensing play a significant role in the improvement of robot performance, vision is recognized as the most powerful of sensory capabilities.

In this paper, we describe a knowledge-based computer-vision system developed to perform the major role in an inspection system which rejects, in real-time, defective products carried by a fast-moving conveyor belt. The paper is organized into six parts. The next section lists some general requirements of the application. The inspection system is then briefly overviewed, and the function of each module composing the system is described. Next, we present an intelligent, knowledge-driven, edge boundary extractor which plays a major part in the system. The algorithms for verification of object structures are described, and finally, some concluding remarks are given.

## REQUIREMENTS

A very sizable proportion of products in the manufacturing industry have round or rectangular shapes. All these manufactured parts and products need to be inspected with reasonably high precision before they are assembled into systems or packaged for distribution. These components or products (which could touch or overlap each other) are normally placed on conveyor belts which carry them to assembly or packaging stations. In many cases, including the food-industry application to be discussed in this paper, the conveyor has its own complexities; for example, a couple of bands running longitudinally and supporting the objects being inspected. Typical requirements for such an inspection system are:

- To perform non-contact automatic inspection at on-line speeds.
- To analyse multiple objects in each frame of the image at on-line speeds.
- To be capable of handling touching objects.
- To perform shape inspection (object size measurement, boundary flaw detection, etc.) and texture analysis, etc.

## OVERVIEW OF THE INSPECTION SYSTEM

To satisfy the real-time requirement and the low system-cost requirement, a transputer-based system has been developed. A CCD camera with a shutter

0952-1976/89/040315-06$2.00

speed up to 1/2000 sec is used, together with a Frame Grabber to acquire a reasonably tidy image (no smearing effect). The image is then passed on to the inspection phase which consists of four major modules. The function of each module is briefly described as follows:

*Intelligent Boundary-Extraction Module (BEM)*
The boundaries are extracted directly from grey-level images using intelligent edge-detection and boundary-following routines. This is an extremely important module because all the subsequent modules use the boundaries of objects as the whole or a part of their working bases. The intelligent, knowledge-driven, edge-boundary follower will be described in detail in the next section.

*Shape-Analysis Module (SAM)*
This analyses the boundary and calculates the size of each object, as well as detecting flaws on the object boundary so that an object of the wrong size, or with defects such as cut corners, nicks or bumps, can be detected.

*Texture-Analysis Module (TAM)*
Burns and items with unacceptable appearance are located by calculating texture coarseness parameters.

*Hole-Detection Module (HDM)*
Because objects with holes inside are unacceptable in this special application, this module detects holes in objects, using thresholding methods.

When the work in each module has been completed, the inspection results is sent to the Graphics Display Module (GDM) where each biscuit outline is displayed on a colour monitor with red colour for failed and green for passed. At the same time, in the practical situation, a signal is sent to a controller to control the rejection of defective biscuits.

## BOUNDARY EXTRACTION

Object inspection techniques can be viewed as either region- or boundary-based. Typically, region-based techniques are geared to the detection of flaws in object surfaces, whereas boundary-based techniques are geared to the detection of flaws in object shapes. Many region-based techniques, however (such as the texture-analysis and hole-detection methods used in our inspection system) assume that the boundaries of objects are available. Before describing the boundary-inspection algorithms, we first of all briefly present our intelligent boundary extractor. (The details of its development are given in Reference 1.)

In addition to many common considerations in designing an efficient and reliable boundary follower, two further problems must be solved to enable good boundaries of objects to be extracted in the class of applications which includes that described in this paper. The first one is that the objects might touch or overlap each other. This causes false information to be passed to the high-level processing system, and therefore causes ambiguity. To distinguish objects from the background detail is another problem, since the grey-level of the objects is very similar to that of the conveyor system. *Figure 1* shows an image captured from the line. The boundaries extracted using a conventional boundary follower[1] are shown in *Figure 2.* It is obvious that it would never be possible for a high-level system to make correct decisions based on these boundaries. Of importance is also that, in designing a satisfactory boundary-extraction system, one cannot forget the industrial realities—the situation is clearly an industrial one, and because each frame of the picture must be analysed within a fraction of a second, this clearly does not allow for sophisticated operations for separating the touching objects. Thus, any lengthy and non-deterministic operations must be avoided.

Therefore, in designing extraction algorithms for industrial applications, we need appropriate strategies to avoid any lengthy and non-deterministic operations. To achieve this, incorporation of

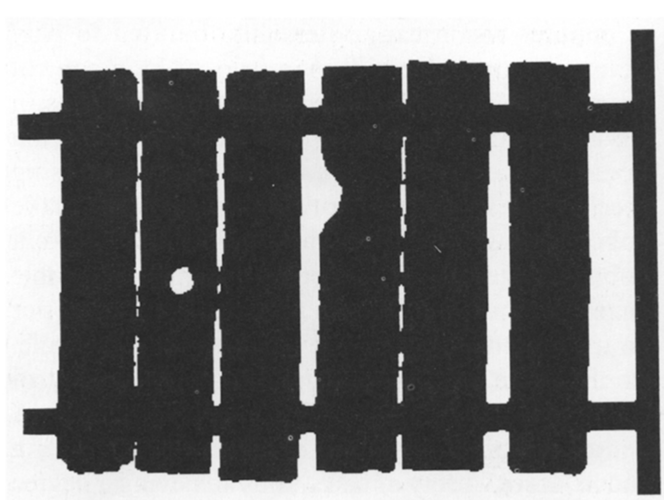

*Figure 1* A frame of the picture captured on-line

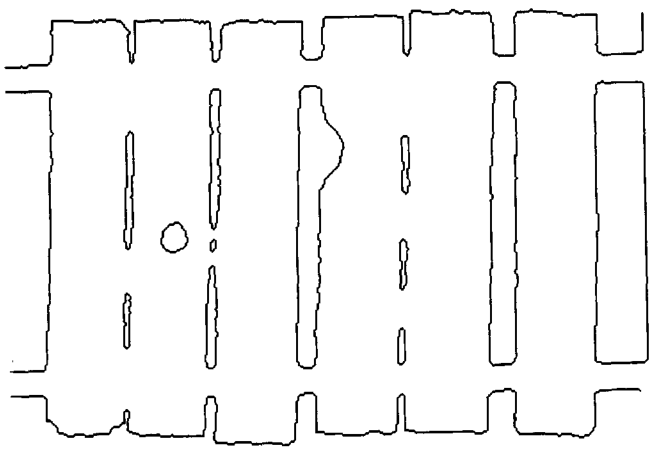

*Figure 2* Boundaries extracted using a conventional boundary follower

'knowledge' in the boundary-extraction system—of, say, the background, the nature of objects to be inspected, and any other necessary information—is essential. In essence, it is important to view boundary-extraction algorithms not simply as mathematical procedures, but as closely linked to the environment in which they operate.

To resolve all these problems, we have designed an integrated, knowledge-driven and intelligent edge-boundary extractor. The details are described in Reference 1. A brief review is given below.

A fast boundary follower directly working on grey-level images was developed for non-touching objects. Unlike many global thresholding and binarization schemes, it thresholds and binarizes images only at pixels where this is necessary. Thus the time taken for tracing boundaries of objects in a grey-level image is no more than the time taken in a binarized image. The boundary-tracing procedure developed is based on the one described in Reference 2. Its kernel is described as follows:

Assume that the image resolution is $M \times N$, then in the image $I(0 \leqslant x \leqslant M,\ 0 \leqslant y \leqslant N$: $x$ and $y$ integers$) \in (0, \ldots, 255)$, let $(x_0, y_0)$ be an arbitrary contour element ($I(x_0, y_0) \leqslant T$: $T$ a predefined threshold) of an object in the scene, and $(x_{b1}, y_{b1})$ be an arbitrary background element ($I(x_{b1}, y_{b1}) > T$) is in the 8-connectedness of $(x_0, y_0)$. To find the next edge point, the elements are searched clockwise, starting from $(x_{b1}, y_{b1})$, in the 8-connectedness of $(x_0, y_0)$, and in the first object element ($I(x, y) \leqslant T$) found is named $(x_1, y_1)$. Then the search for object elements (in the neighbourhood of $(x_1, y_1)$) is resumed, clockwise, from $(x_0, y_0)$, and $(x_2, y_2)(\neq (x_0, y_0))$ is then chosen in the same way as $(x_1, y_1)$. This procedure is repeated, and the sequence of contour elements $(x_3, y_3), \ldots, (x_N, y_N)$ ($N$ is the number of boundary points) is determined successively until $(x_N, y_N) = (x_0, y_0)$ and $(x_{N+1}, y_{N+1}) = (x_1, y_1)$ hold for the first time. Thus a pair of coordinates of each boundary point of an object is directly extracted as $(x_0, y_0), \ldots, (x_i, y_i), \ldots, (x_N, y_N)$.

Armed with this very effective method of boundary extraction for ideal, but grey-level images, we now can approach the next set of problems posed by industrial environments! This is to build the 'intelligence' into the boundary extractor, using available knowledge. Some useful *a priori* knowledge for boundary extraction in this application includes:

- Objects are placed on the conveyor belt in the direction shown in *Figure 1* with only slight orientation changes.
- Touching between objects occurs only in small areas, due to the coarse edges of biscuits.
- All objects have rectangular shapes, therefore the edges of objects are straight lines.
- The width of each band of the conveyor system is far shorter than the biscuit length.

After extensive investigation, it was found that a boundary follower, which is efficient in terms of high speed requirement and has intelligence to resolve ambiguities caused by touching points, etc., must integrate routines with implicit rules to do fast boundary tracing and with an explicit rule-based system which 'knows' when a touching point is met. It then should be able to make decisions to tell the follower which way to go. However, it must be pointed out that the explicit rules used for separating, say, touching objects will normally be different for different objects. Thus such a scheme designed for a particular application will be largely inapplicable to others. A means of overcoming these problems is to have available a large set of segmenting rules, some of which can be used for specific situations. However, only certain rules will be needed in each case, and thus a software structure is required to manage the rule usage.

The same principles also apply in the design of efficient routines for locating objects and to routines for eliminating non-significant features—such as those observed in background scenes.

The first step in extracting the boundaries of any object in an image, using the contour follower just described is to locate an object and find a point on each object from where to commence boundary tracing. As has been discussed in Reference 1, unintelligent operators, such as run-checking,

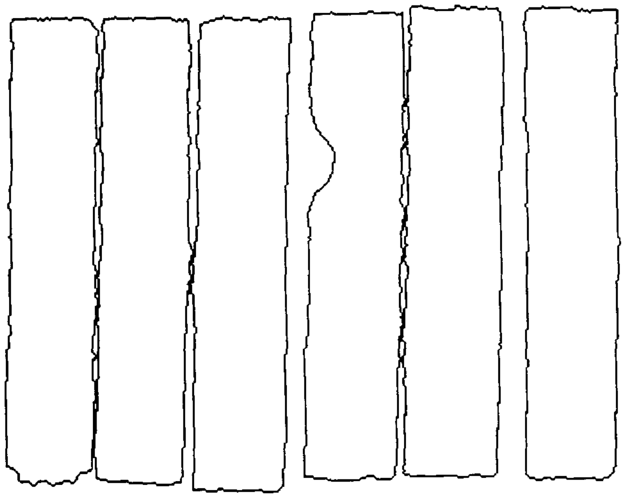

*Figure 3* Boundaries extracted using the intelligent boundary extractor

cannot cope with objects touching, or with 'false features' in an image, neither are they capable of eliminating unnecessary operations. In practical situations, the use of *a priori* knowledge about the environment can significantly speed up the object location process. For example, if the objects to be inspected are supported by, say, a double-band conveyor system, the process adopted for locating objects in an image can be reduced from scanning the whole image to just one line, say, the middle line of the image. This is feasible as the middle line crosses all the objects in the image.

This idea of simplifying the search area, is a very practical solution and has been exploited elsewhere. For example, in the work by Berger[3] on rock-size measurement, it was found that the conveyor system was slightly concave. As a result, all objects on the belt tended, to a high degree, to be approximately in the centre, or mid-line, of the belt. A procedure, referred to as the *midline scheme* can thus be introduced to locate the objects (details are described in Reference 1).

The basic principle suggested for eliminating false feature effects, appropriate to the particular industrial applications tackled in this paper, is that of knowledge acquisition. Before the inspection starts, a program is run to allow a frame of the image of the empty belt to be captured, so that the position of the belt and its features can be stored by the system, and this information is subsequently regularly checked during on-line inspection. During the boundary-tracing phase, the tracer is given the intelligence to know if it hits, say, the belt's edge. If it does, it 'jumps over' the belt and resumes the tracing, and this part of the object boundary is replaced by a straight-line segment. This procedure is, of course, application-dependent, and is acceptable here in that the width of each band of the conveyor is far shorter than the object length, and the edges of the objects are straight lines!

The belt elimination procedure is essentially self-learning. For example, as a boundary is traced, it can learn about the approximate orientation of the objects lying on the belt, and thus it goes to the approximately right position to look for boundary-trace-resumption points.

Thus, by the incorporation of explicit, environmental knowledge, we have solved touching and false feature problems. We have also sped up the boundary extraction process by eliminating lengthy operations using *a priori* knowledge. The overall boundary extraction scheme now is:

1. Locate the major objects in the current frame of the image.
2. Activate the fast boundary follower and start tracing the boundary.
3. If a touching point or false feature is met (i.e., if there are more than two paths to follow) then processing is moved to a knowledge-based routine which makes a decision on which way the boundary should go.

The output of this scheme generates a list of coordinate pairs corresponding to each object boundary. *Figure 3* shows the object boundaries extracted from the image shown in *Figure 1*, using the above intelligent boundary-extraction scheme. As has been shown in Reference 1, the distortion caused by touching is reduced to a minimum—typically 1 pixel—by using the above scheme.

Having investigated the problems relating to boundary extraction in real-world, industrial scenes, we now move on to the problem of boundary verification—a major item for product quality control.

## BOUNDARY ANALYSIS

The purpose of analysing the boundaries of objects is to reject objects with faulty shapes. The techniques developed for this involve following a sequence of four major steps.

### *Detect local maxima on the curvature graph*

Since there are significant changes in edge orientation at corners of an object, the analysis of changes in curvature at boundary points provides

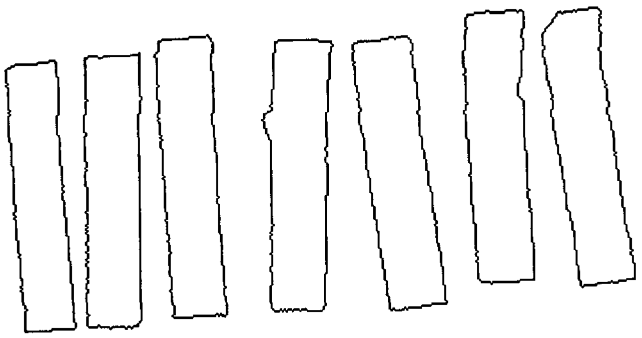

*Figure 4a* The object boundaries

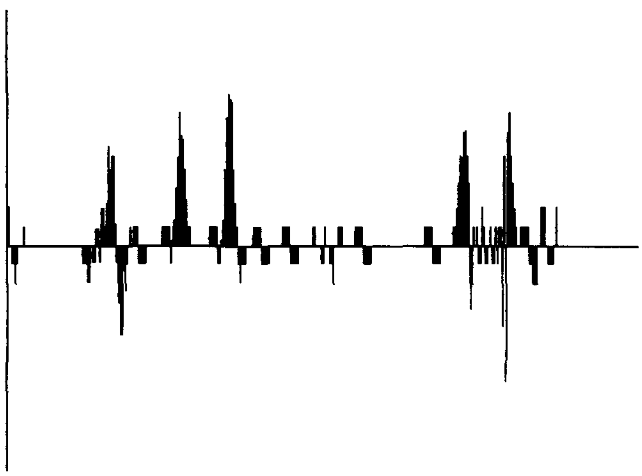

*Figure 4b* The curvature plot for the fourth boundary

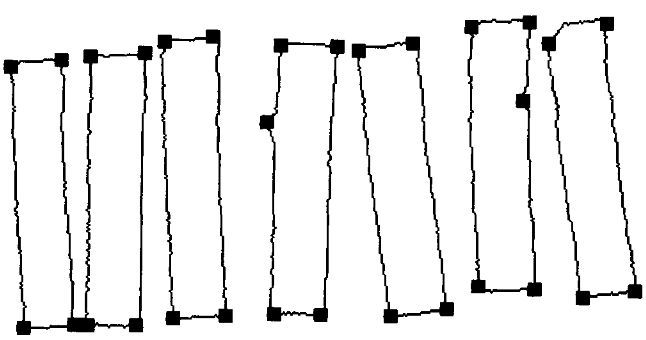

*Figure 4c* Local curvature maxima

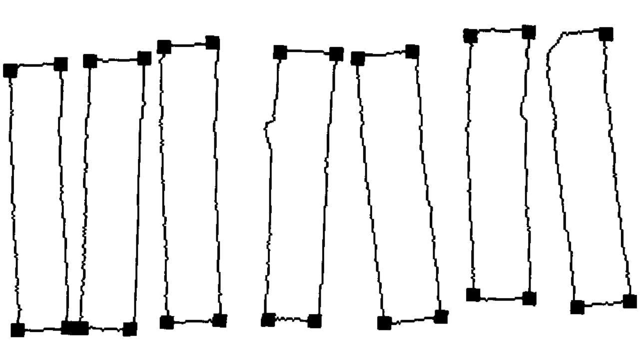

*Figure 4d* Corners detected

useful information for locating the object corners. A simple but efficient algorithm for calculating curvature of a discrete curve can be found in Reference 4. *Figure 4b* shows a curvature plot for the boundary shown in *Figure 4a.* To find the major points of discontinuity, a threshold is applied to the curvature graph, and those points with a curvature value greater than the overall threshold and with a local maximum curvature value, are selected. As a result, only the real corner points and any other points where there are possible defects causing high curvature values, are chosen and passed to the next stage of processing. In *Figure 4c* the points marked with a small square black area are the points selected after this stage.

*Eliminate non-corner points*

To be able to estimate the size (length and width) of an object in the scene, we need to know which points found in step (1) are the 'real' corners, and which ones are not. This can be achieved by estimating the angle between two vectors. Assuming that $P_1, \ldots, P_i, \ldots, P_m$ are the points selected by step (1), whether or not point $P_i$ is a 'real' corner depends on whether the angle between vectors $P_iP_{i-1}$ and $P_iP_{i+1}$ is equal or close to $\pi/2$. *Figure 4d* shows the corners found.

*Calculate the size of each object*

Having obtained the real corners of an object, its size can then be calculated by simply calculating the distance between two adjacent corners. The average of the two longest distances is the object length, and the average of the two shortest is the object width. It should be noted that in some circumstances, only three corners can be found, in which case the object is simply rejected, since a cut corner exists.

*Detection of ragged edges*

There are two ways of inspecting the boundary conditions. One is to calculate the distance between each boundary point and the straight line formed by two adjacent corners. The other is to calculate the distance between every boundary point and a straight line obtained by using the information on all boundary points between the two adjacent corners. It is clear that the former method is fast but not very accurate, and the latter is just the contrary. If an object fails either of these distance tests (i.e., a distance is greater than the allowed tolerance), it is rejected. In practice, if this case is met, some other measurements regarding the condition of the boundary are also made, using available knowledge.

After the shape analysis, only objects which have passed the shape test are passed to other inspection modules, although currently all objects are going through all inspection modules in parallel for the convenience of the parallel implementation of the whole system on a transputer network.

## RESULTS AND CONCLUSIONS

The system has been tested on a scale-model production line. The results show that the system developed is robust and flexible, and can provide real-time performance. The inspection results show that correct classification rate is high and classification for the same frame of the image is repeatable. In the case of touching objects, the intelligent boundary extractor can generate reasonably good boundaries under good back-lighting conditions. The time taken for analysing (including all the necessary processing) each frame of the image is normally less than 0.5 seconds, and therefore directly matches the on-line requirements. As the system has a modular structure, it can easily be improved by replacing a particular module by one which uses better inspection strategies.

## ACKNOWLEDGEMENT

The work on texture analysis and hole detection was performed by J. Powrie, and the system integration was completed by the authors, together with J. Powrie and F. Deravi. That work is to be reported elsewhere. We wish to thank J. Powrie, F. Deravi and X. Cao for helpful discussions during the course of performing the work described in this paper.

## REFERENCES

1 Wu, Q. M. and Rodd, M. G. *Fast Intelligent Boundary Extraction of Touching Objects.* Internal report, University College of Swansea (January 1989)

2 Yokio, S. et al. An analysis of topological properties of digitized binary pictures using local features. *Computer Graphics and Image Processing*, **4** (1), 63–73 (1975)

3 Berger, G. F. *Rock size measurement.* Dissertation, University of Witwatersrand, South Africa (1986)

4 Wu, Q. M. and Rodd, M. G. Boundary segmentation and parameter estimation for industrial inspection. *IEE Proceedings-E* (March 1990)