Real time machine vision inspection of plastic bottle closures

Trevor C. Bartleet Gerhard de Jager

University of Cape Town, Department of Electrical Engineering

ABSTRACT

This paper presents a successful implementation of a real time inspection system of plastic bottle closures. The closures are inspected at a rate of 20 per second or one every $50ms$. The available time to inspect each closure forces the algorithms used to be relatively simple. Even though the algorithms are simple,they have to be robust to ensure a good result. Two boundary tracking algorithms were designed and implemented, one based on edge strength information and one based on threshold information. Both use prior knowledge about the closure. The results are sufficiently accurate to replace a human operator with the machine inspection system. A description of the system including the timing and hardware used is given. The results achieved using each of the algorithms are presented. This is followed by a brief discussion of problems associated with achieving highly accurate results using a simple algorithms.

Keywords: real time, machine inspection, boundary tracker

1 INTRODUCTION

The inspection of plastic bottle closures for defects carried out by human inspectors is an unrewarding task. The inspection process is subjective and it is difficult to see many of the defects. A machine vision system has the advantage of being able to inspect every closure and yield consistent results despite an environment that is unpleasant for humans to work in.

Plastic bottle closures or bottle tops are moulded and then lined at approximately 20 per second. The liner placed in these closures may be defective. Quality control of the final product, to levels of fractions of a percent, requires the inspection of every closure and subsequent removal of defective closures.

If a significant number of defects are produced, the lining machine needs adjusting to eliminate the problem. A machine vision system can inspect every closure and provide feedback as to the number and type of defect being produced. This feedback can be used in the monitoring and adjusting of the production, thus ensuring final quality control with as little wastage as possible.

This paper presents the development and implementation of such a machine vision system. On the production line in the factory, where it is installed, the system produces sufficiently good results to replace the human inspectors and achieve a better result. The overall design of the system as a whole is presented in part 2. This includes the lighting, an important aspect, as well as the method used to detect the position of the closures in the system and to eject the defective closures.

The algorithms used to find the defects are discussed in part 3. Each image has to be processed in real time, that is, in less than $50ms$. Based on this time constraint, two different algorithms have been developed and tested. Each extracts the boundary of the liner, this gives information about its shape and position within the image. The way the shape and the pixel intensity of the boundary vary determines if there is a defect in the liner or not. This is described in part 4.

Part 5 presents the results obtained for these algorithms when installed on the production line. The accuracy or effectiveness of the machine vision system is discussed.

2 SYSTEM DESCRIPTION

The machine vision inspection system consists of four principle components: the host computer which manages the system and the user interface; the DSP for processing the images; the frame-grabber to digitise the images and the devices which interact with the environment, the CCD camera, closure detector and closure ejector. The interaction between the different components of the system are shown in Figure 1 and a brief description of each of the major components is given in the following sections.

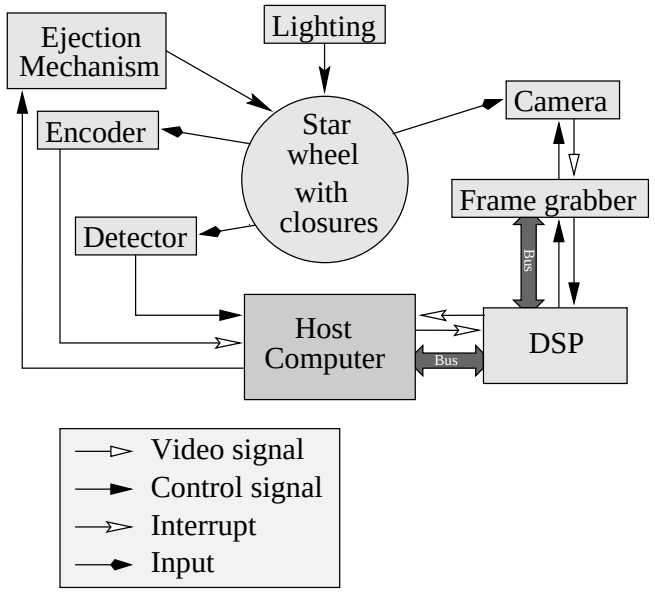


Figure 1: Interaction of the different parts of the system.

2.1 Host computer

The host computer is responsible for managing the system and running the user interface. It also receives and manages the information from the transducers on the production line.

The management of the system involves various timing considerations. For every closure this involves:

- The detection of the closure that has just been produced.

- Signalling the camera to take and image and simultaneously instructing the frame-grabber to digitise the image.
- Signalling the DSP to process the image.
- Receiving the results of the processing from the DSP.
- The ejection of the closure at the correct time if necessary.

Timing has to be maintained for each and every closure passing through the system.

The host computer also checks for fatal conditions on the DSP. If it fails to respond for too long it is reset and the code is loaded onto the DSP. If a result arrives from the DSP after the closure has passed the ejector, the result must be ignored.

The user interface runs on the host computer and allows the user to configure the system. The results of the inspection are logged and statistics are generated. From this alarms can be set to provide feedback to the system operators.

2.2 Frame-grabber

The image from the camera is sampled by a frame-grabber which shares the same bus as the DSP processor. In this way, when the image is stored in one of the frame buffers as it is digitised, it is ready to be processed.

2.3 DSP processor board

The DSP processor used is a Motorola 56001 running at $33MHz$. The image memory is divided into two frame buffers each capable of holding one image.

The signal for an image from the camera takes about $50ms$ to be digitised. While the image is digitised into one frame buffer, the image in the second buffer is processed. This ensures that the time is effectively utilised.

As the closures pass through the system at approximately 20 per second, the time available to process any single closure is $50ms$. Any algorithm must therefore take less than $50ms$ to execute the entire algorithm and to return the result of the processing to the host computer.

2.4 Host computer - DSP processor communication

The host computer and the DSP run asynchronously. As a result an asynchronous communication system has been developed so that the host computer can send instructions to the DSP and the DSP can return the results of its process to the host computer. Classification of the closure can then be done and the closure accepted or rejected. The host computer has to keep track of the closure to which the result from the DSP belongs. Care has to be taken that the communication does not lock up between the two processors.

2.5 Lighting

It is well known that the correct use of lighting can simplify image processing problems. With this system, good illumination was difficult due to the size of the closure and its proximity the lens. The lens is positioned close to the closure in order to use the distortion of the lens to make the defects of the liner more visible. A small gap between the top of the liner and the bottom of the lens allows sufficient light into the closure. The amount of light determines the aperture setting of the lens and the possible shutter speed. The size of the aperture determines the depth of field:- to get a good depth of field, the aperture needs to be as small as possible. On the other hand to get the shutter speed fast enough the aperture needs to be open to let enough light in for the image formation.

A ring of 24 LEDs was used to illuminate the closure. The light was diffused using frosted glass to get the light as uniform as possible around the boundary of the liner. With this arrangement the shutter speed could be set at $\frac{1}{4000}s$,fast enough to prevent blurring of the image due to the motion of the closure.

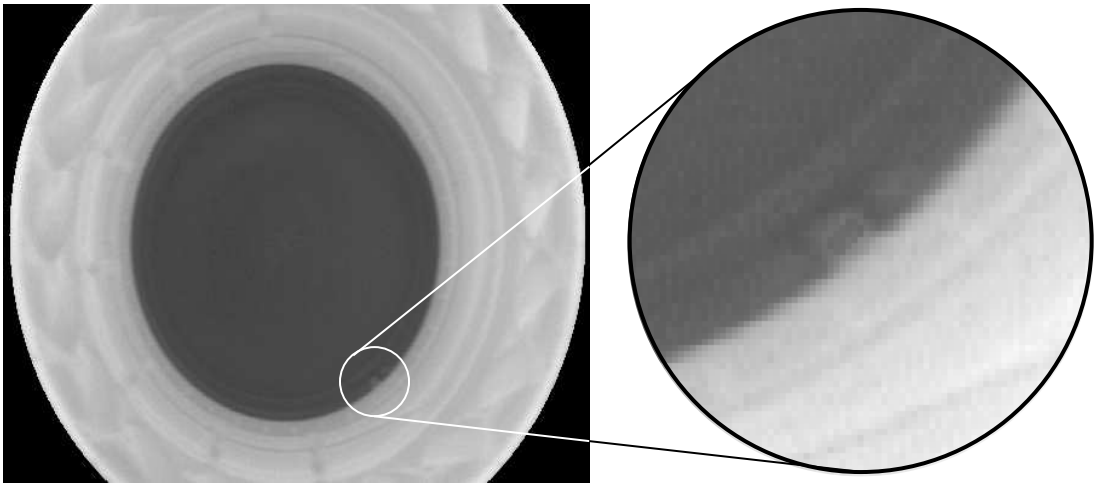
3 ALGORITHM CONCEPTS

The two defects that are being searched for both occur along the boundary of the closure liner. Two examples of these defects are shown in Figure 2. The design of an algorithm to solve this problem has a time constraint of $50ms$. This limits the complexity of and the possible amount of image information processed. The design of any algorithm must access as few pixels as possible, thereby reducing the amount of image information being processed and the complexity of the processing. Another way of reducing the complexity of processing is by using prior knowledge about the closure.

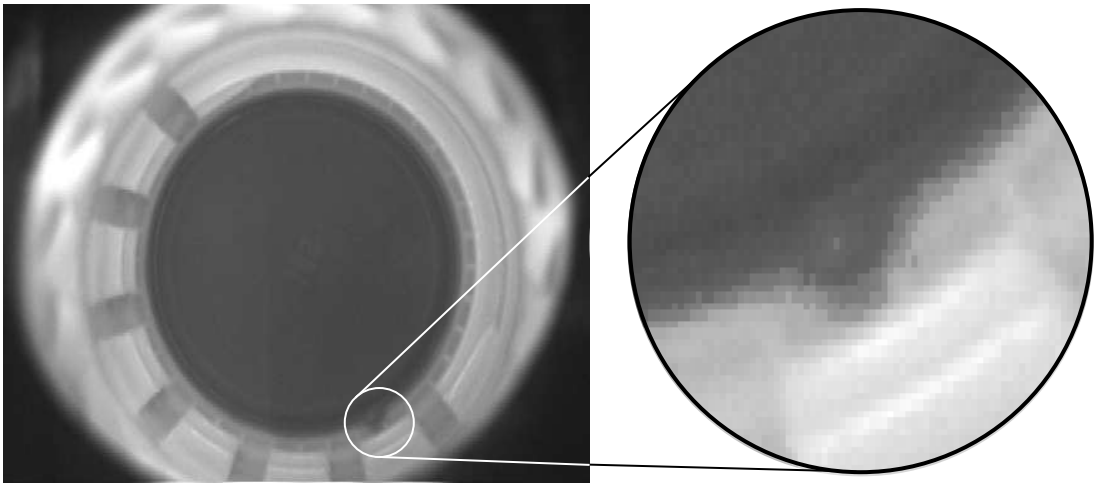
The defects under discussion occur near the edge of the liner. The problem therefore involves finding the boundary of the liner and examining the shape and the pixel intensities along the boundary. Taking the above factors into consideration, a boundary tracking algorithm was chosen to locate the liner boundary. This algorithm calculates the shape of the liner boundary, its position and the grey level intensities along the liner boundary. This information is sufficient to make a decision as to whether there is a defect in the closure liner or not. The advantages of a boundary tracker are: (a) that it can be designed to use prior information about the shape of the boundary that is being searched for, and (b) the boundary tracking algorithm only uses local image information. With the boundary tracker it is simple to tell if processing is complete:- the track returns to its starting point. Two different boundary trackers where implemented, the first used edge strength information and the second used threshold information.

3.1 Boundary tracking based on edge strength

The closure liner is much darker than the closure and there is an abrupt change in the intensity of the image between the two regions. This results in a strong edge being present in the image. This edge will usually run in a circle, the shape of the liner. Thus from a know starting point the direction of the edge is known and it is possible to choose a simple edge detector which is able to find the best edge for each direction of the liner edge. The edge between the liner and the closure is a rapid transition, therefore the edge detector only needs to examine pixels in a small neighbourhood of the current edge pixel. The boundary found can then be used to determine if the liner has any defects.



(a) Non-fill



(b) Flashing

Figure 2: Closures showing two possible defects.

3.2 Boundary tracking based on threshold information

This method relies on the good contrast between the liner and the closure. As the lighting is uniform, there is no slope across the liner so a single threshold can be chosen to separate the liner from the closure. Haralick and Shapiro[1] describe a usual method of using a histogram to decide on a threshold value to separate the background and the foreground in an image. This method is not suitable here and the solution to the problem is to use prior knowledge about the position of the closure in the image and simply sample the intensity in a region. The threshold level can be set slightly higher than the sampled level of the liner. Once this threshold level has been chosen, a binary boundary tracker can be used to follow the outside of the liner. This boundary can be used to determine if the liner has any defects.

3.3 Different effects achieved by the boundary trackers

The two boundary trackers may appear to be very similar methods of determining the liner boundary, however they differ on a fundamental issue. The method based on edge strength simultaneously attempts to extract the boundary and at the same time determine its position. This has two disadvantages: the first is that there are other areas in the image which have strong edges but these edges are not the edge of the liner. These strong edges are found on the thread of the closure. The second problem is that edge detection is a differentiation operation and thus is susceptible to noise. The resultant boundary location can be incorrect and any classification done on this boundary information will not be accurate.

The threshold method first extracts the liner using the threshold and the boundary tracker then determines the position. This method is better as the segmentation is done separately, before determining the boundary position. The weakness of the threshold method is that the threshold level is determined using a sample of only a few points. If the threshold level is not correctly chosen, then the liner is not correctly segmented. This method is not as susceptible to noise as the image is not differentiated and the area is averaged which lessens the effect of noise.

4 DETAILS OF IMPLEMENTATION

This section examines points particular to the implementation of the the algorithm on the DSP. The implementation of the boundary tracking algorithms and the algorithms which process the output of the boundary tracker to determine characteristics of the liner boundary to be used to decide if there are defects or not.

4.1 Boundary tracker using edge strength

The first part of the algorithm is to find a starting point on the liner boundary. Since the approximate size and position of the closure is known, the area where the start could be is limited. The start is defined as the strongest edge on the centre row of the image. Once the starting point has been found, tracking can begin.

The boundary is divided up into 8 directions, these are the possible directions that the boundary can take. The edge detector used in any one of the 8 directions is perpendicular to the edge of the liner. At each boundary position, three possible pixels are examined to determine which is next. These are the pixels in the current direction of the boundary and one either side of it. The edge strength at each of these positions is examined and the next position is at the pixel where the edge strength is a maximum. The past history of the boundary is examined at each move and if the slope of the boundary has changed sufficiently, a new principle direction

is chosen. The track stops when the current row is the same as the start row. This boundary tracker is not guaranteed to finish at the same position that it started.

4.2 Boundary tracker using threshold information

This boundary tracker assumes that the boundary being searched for remains within the image, this will be the case if the threshold is selected correctly. The area over which the start position is searched for is limited by prior knowledge about the size and positioning of the closure in the image. The start is searched for from outside of the closure until the first pixel below the set threshold is found:- this is the starting point.

The boundary tracker moves in a clockwise direction around the liner. This is done by examining a maximum of 8 neighbours of the current pixel. The track is complete when it gets back to the starting pixel. Checks can be made based on the length of the track to decide if the track could be acceptable. If the track is too short or too long the track cannot be considered to be good.

4.3 Looking for flashing

Once the liner boundary has been found, it needs to be examined to determine if features indicating flashing, are present. As the expected shape of the liner boundary is circular and flashing should cause a deviation from this, a measure of symmetry has been determined to indicate the deviation from the ideal shape. This is done by determining how well the two halves of the boundary match. The difference between the boundary at the top and bottom of the boundary is determined and then the difference in the boundary on the left and right half of the image.

4.4 Looking for non-fills

Non-fills occur along the boundary of the liner. They appear in the image as spots slightly darker than the rest of the liner. The method used to find these non-fills involves using an intensity profile along the boundary. This intensity profile is determined by taking lowest pixel value of either the boundary point or two pixels inside the boundary. Inside is defined as the shape of the boundary is known.

A non-fill will have a signature in this intensity profile, where the non-fill is along the boundary the intensity profile will be lower than the rest of the signal. This low section in the intensity profile is searched for by finding the maximum variance over a window. The window size is not critical but it must be sufficiently small to not be affected by gradual changes in the intensity profile. Figure 3 shows a typical intensity profile with a non-fill.

5 RESULTS

The results were generated on the production line on runs of between 10 000 and 20 000 closures. The results for the system using the boundary tracking method based on edge strength information is shown in Table 1 and the results for the boundary tracker based on the threshold information are shown in Table 2. These tables show the three classes of closures as rows and the columns show the actual classification of each of these classes.

The edge strength boundary tracker correctly classified approximately 99% of the good closures and 80% of

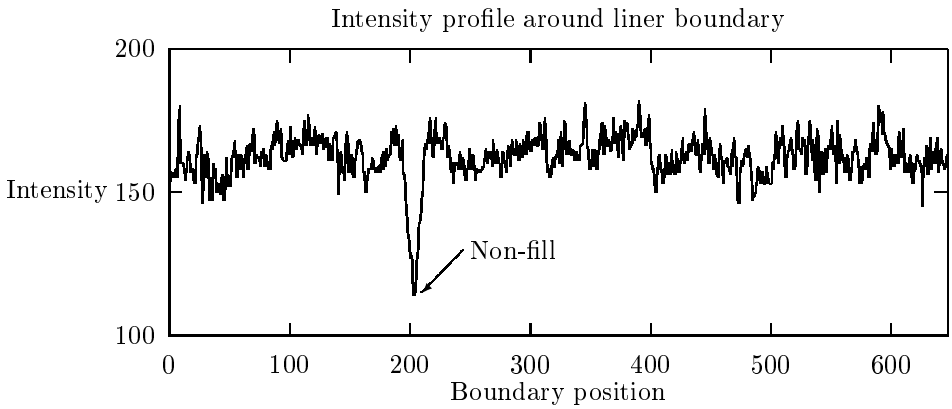


Figure 3: Intensity profile along the liner boundary.

the defective closures correctly. The 1% of good closures that are incorrectly identified are ejected. This is quite high as the total number of closures rejected a day yields a substantial amount of unnecessary waste. There are approximately 2 000 000 closures produced, 1% of this means 20 000 closures being wasted. The 80% rejection of defective closures is not good enough. Changing the threshold levels at which the features are separated does not have a large effect. Good closures may be incorrectly classified because of noise or other effects creating the appearance of a defect. Defects in closures that are not identified occurs when the algorithm does not detect the defects and a correct classification cannot be made.

The threshold boundary tracker achieves a much better result. The average classification was approximately 99.9% of the good closures were correctly classified. This means only 0.1% of closures were wasted, or about 2 000 a day. The correct classification of non-fills remains at about 90% and the correct classification of flashing improved to more than 90%. Again changing the thresholds in the classification algorithms does not improve the results. The defective closures are missed due to the algorithm not detecting the defects.

	Classification		
	Good	Flashing	Non-fill
Good	99.26 ± 0.38	0.42 ± 0.10	0.33 ± 0.28
Flashing	21.05 ± 0.72	78.96 ± 0.72	0.00
Non-fill	16.60 ±	0.00	83.40 ± 5.62

Table 1: Results of the boundary tracking algorithm based on edge strength information.

	Classification		
	Good	Flashing	Non-fill
Good	99.84 ± 0.12	0.26 ± 0.12	0.00
Flashing	8.69 ± 6.37	91.30 ± 6.37	0.00
Non-fill	7.79 ± 7.79	0.00	92.21 ± 7.79

Table 2: Results of the boundary tracking algorithm based on the threshold information.

6 DISCUSSIONS AND CONCLUSIONS

6.1 Factors influencing the final accuracy of the system

Design of a system to inspect the bottle closures at 20 per second in real time is possible. The algorithms used are simplistic due to the time constraint. This prevents the rigorous checking of various influences on the image. These include the tilting of the closure under the camera causing slight blurring, the cap moving slightly closer to the camera again causing blurring, shiny spots on the liner caused by water in the compressed air used to punch the liner into the closure, amongst others. All these external influences can disturb the image causing incorrect classifications.

To obtain a correct result on most closures is a relatively simple task. To improve these results to be more accurate is difficult without looking at more information and taking more features into account. There is a limit as to how well a simple algorithm can work based on the large number of possible variations in the final image. This limits the overall accuracy of the final system.

6.2 Conclusions

A machine vision inspection system is described that has been implemented in real time with $50ms$ per closure being used for processing. The results obtained are sufficiently accurate to implement the inspection system.

7 REFERENCES

[1] Robert M. Haralick and Linda G. Shapiro. *Computer and Robot Vision*, volume 1. Addison-Wesley Publishing Company, 1992.