# Real-time Automatic Machine Inspection of Plastic Bottle Closures



Trevor Clive Bartleet

Submitted to the Faculty of Engineering, University of Cape Town, in partial fulfilment of the requirements for the degree Master of Science in Engineering.
Cape Town,
September 1995

# Declaration

I declare that this dissertation is my own work. It is being submitted for the Master of Science in Engineering at the University of Cape Town. It has not been submitted before for any degree or examination at this or any other university.

Signed by candidate

T.C. Bartleet

# Acknowledgements

I would like to thank the following people and institutions for their contribution towards this thesis.

Professor Gerhard de Jager for his guidance and help.

My Mom and Dad for their support and encouragement.

Henri Johnson, Bob Rust and Theo Lindebaum of Electronic Development House for their assistance during the project.

Fred Hoare for administrating the computer resources.

The Foundation for Research and Development for their financial support.

The digital image processing group.

# Abstract

*This thesis presents work done developing an automatic inspection system for plastic bottle closures. These plastic closures are manufactured at 20 per second and the inspection must be done in real-time at this rate. Two types of defects must be detected and the defective closures removed from the production line.*

*Inspection by human operators proved to be unreliable and subjective, and an automatic inspection system was needed. A machine vision system is capable of producing reliable and repeatable results in an environment which is unpleasant to humans. An automatic inspection system was implemented on the production line.*

*The development of this system is divided into three sections. The first is the design and selection of the hardware. This includes the camera, digital signal processing board, the host computer and transducers which interact with the environment.*

*The second is the development of the image processing algorithms used to detect defects in the bottle closures. Two methods were attempted to do the image processing. The one is based on the branch of image processing known as mathematical morphology. These algorithms were not implemented on the digital signal processor as they could not be implemented in real-time. The other method used was to track the boundary of the liner and use information from this to detect the defects. This method was implemented in the industrial system.*

*The third was the development of the software for the host computer which manages the overall system. The management includes: a user interface, communication with the digital signal processor in order to receive the results of the inspection, the interaction with the environment, and the overall timing within the system.*

*Results have been obtained where 99.9% of the good closures are accepted, and more than 90% of the defective closures are removed from the production line.*

# Contents

# List of Figures

# List of Tables

# Chapter 1

# Introduction

## 1.1 Background

Plastic bottle tops or closures are produced for the bottling industry. During summer when the local demand for soft-drinks is high, most of the factory's output is used to supply the needs of the local market. In winter when the local demand is low, most of the factory's production is exported to Europe and the Middle East.

During the production of the closures a number of defects can occur. These defects cause the bottles to leak and gas to escape. In order for the closures to be acceptable, to the bottling companies, they need to be of a high quality with a low defect count. This means that the quality control of the closures needs to be done to levels of a fraction of one percent. This involves the individual inspection of every closure.

Inspection of the closures for these defects carried out by human inspectors is an unrewarding task. The environment is noisy and unpleasant to work in and the human attention span is limited. Results are therefore inefficient and subjective. The overall cost of the production of the closures is increased through the wages paid to inspection personnel. These are all fundamental problems with the manual inspection system, however, the most important factor is the manufacturer does not know exactly how many defective closures are being shipped to the consumer. This means the manufacturer does not have strong control of the inspection process.

In order to produce a high quality product at a reduced cost an automatic inspection system is required.

## 1.2 A machine inspection solution

Industrial inspection is an area where the use of machine vision technology is eminently suitable. Areas where automatic machine vision inspection systems have been used are the inspection of wafers used in semiconductor manufacture, Yoda *et al* [26]. Hara *et al* [11] developed a printed circuit board pattern inspector. The inspection took 18 minutes for a $500 \times 600$ $mm$ surface. Boerner *et al* [1] used an automatic x-ray inspection of light-alloy pieces used in the car industry. Human inspection takes about 90 seconds per part. Boerner states that processing time is longer than for a human operator but is in a tolerable range. These automatic inspection systems all inspect reasonably complex objects. These are computationally intensive and the inspection involves the whole image.

Shabushnig [23] used a line scan camera to inspect the label position on a cylindrical bottle and whether a child-resistant closure was correctly applied to the bottle. Inspection for both the label positioning could be done at up to 110 feet per minute. Inspection of the child-resistant closure application was done at up to 91 feet per minute. Thomas and Cattoen [24] developed an automatic inspection system for plain and simply patterned material passing at several tens of meters per minute.

The inspection of bottle closures requires the detection of defects in a closure. A closure may be considered as a simple object as it has a fixed shape, a simple structure and only two colours, white and grey. This means the computational requirement is much less than that required for more complex objects and indicates that it is possible to develop an automatic machine inspection system to inspect the closures. A machine inspection system must have the following characteristics

- Every closure must be individually inspected to determine if there is a defect.
- Must work in the factory environment 24 hours a day.
- Must yield consistent, repeatable results.
- There must be a maximum detection of defective closures with a minimum number of good closures being falsely detected as defective.
- Defective closures must be ejected from the production line.
- The reliability of the system must be good.
- Maintenance of the system must be low.

This thesis describes the development of a real-time machine inspection system to inspect

the closures. The inspection is done at 20 per second or one closure every 50 $ms$. The development of the system can be divided into four sections.

- Firstly, the hardware used. This includes: the optical components for image acquisition; the DSP for the image processing; the host computer to control the overall system; the camera and frame grabber used to digitise the image and the transducers used for the interaction of the system and the machine which produces the bottle closures.
- Secondly, a number of different algorithms were developed. Two of these algorithms were implemented on the DSP. The other algorithms were too computationally intensive to be implemented.
- Thirdly, design and implementation of the overall system to inspect the closures. This system combines all the individual components, the hardware, the image processing algorithms and the software managing the system.
- Finally, results generated from the system installed on the production line are presented.

## 1.3 Outline

This section gives a breakdown of the work covered in this dissertation. It can be divided into five sections. Firstly, the description of the problem in detail. Secondly, a review of the relevant theory and how it is used. Thirdly, a discussion of the development of the image processing algorithms used to inspect the closures. This includes a discussion of the use, implementation and effectiveness of each. Fourthly, the implementation of the overall system. Finally, the results of the inspection of the closures are presented.

The following is a detailed breakdown of the work covered in each chapter.

Chapter 2 gives a detailed description of the problem. There are three main sections presented. Firstly, an introduction to the manufacturing process. This includes a description of the different types of closures produced. Secondly, examples of the different defects that can be introduced during the manufacturing process are shown. Thirdly, the specification of the results required for the inspection system.

Chapter 3 is on optical theory and implementation. This chapter gives an overview of some theory associated with the optical components of the system and the formation of the image. The characteristics and influence of the different facets of the system are discussed. This includes blur due to the motion of the closure relative to the camera. The motivation for the type of components and the settings used in the final system are given.

### Section 1.3: Outline

Chapter 4 presents the image processing theory relevant to the work done. The basic theory for the different image processing algorithms that are designed later are given. This includes edge detection and thresholding as well as both binary and grey-scale morphology.

Chapter 5 discusses the hardware used in the system. The hardware used to implement the machine vision solution is very important to the choice of algorithms used. This chapter describes the different hardware components and how they were designed.

Chapter 6 is on image processing algorithms. One of the problems when designing a real-time system is that the algorithms have to run fast enough, to return a result before the next object is ready to be processed. To inspect closures at a fast rate the algorithms chosen are relatively simple. Two different classes of algorithms were used. The first class was based on boundary tracking methods to extract the liner boundary in the image. The second class of algorithms used morphological image processing methods. The characteristics and the success of each of the algorithms are discussed and compared.

Chapter 7 gives the details regarding the implementation of the system. The overall system that was designed is presented. This includes the software which handles the timing implemented on the host computer. The image processing algorithms were implemented on the production line.

Chapter 8 presents the results of the inspection of the bottle closures on the production line.

Chapter 9 gives an overview of what has been achieved as well as further work that is required.

# Chapter 2

# Problem Description



## 2.1 Introduction

This chapter begins by giving a description of the manufacturing process, Section 2.2, which gives the background to the problem. Section 2.3 examines the two types of closures produced. The characteristics of the closures are presented. These characteristics are important as they affect the overall design of the system. Section 2.4 examines the defects which can occur in the closures during the manufacturing process. The nature of the two defects which need to be detected are examined. Section 2.5 specifies the results needed by the manufacturers for an automatic inspection system. Section 2.6 proposes a solution.

## 2.2 Description of the manufacturing process

Plastic bottle closures or bottle tops are produced for use by bottling companies. The closure is first moulded at a rate of ten per second on two moulding machines, these closures are then allowed to cool. Once the closures have cooled, the seal, called a liner, is placed in the closure. This liner is usually made out of grey PVC but sometimes other non-PVC plastics such as Duraform are used. The liner colour is always grey although the shade of grey may vary considerably. The closures are lined at twenty per second.

In the moulding process very few defects are formed once the moulding machine is correctly set up. The lining process on the other hand can cause a number of defects. The number of defective closures produced depends on a number of factors. These include

- The type of lining material used, non-PVC plastics are more prone to cause defects than

PVC.

- Fluctuations in the pressure of the lining material as it is extruded from a jet from the heater which melts the lining material.
- Impurities in the lining material.
- Mechanical imperfections causing the misalignment of the closure being lined.
- Damage to the punches that form the liner, in the closure, from the pellet of lining material.
- The incorrect size of the pellet of lining material used.
- Lack of lubricant in the lining material.

Defective closures must be removed from the production line to ensure a good quality end product. Presently the closures are sorted manually. The single stream of twenty closures per second is divided into four parallel streams running at five closures per second. These closures are inspected by one person. This system is not effective as many defective closures are missed as the inspector loses concentration. Defects are not easily visible as it is difficult to see inside a closure. The only real benefit of the human inspector is the versatility; they are able to make distinctions between a greater variety of defects. On the other hand, a human inspector may be subjective and different inspectors will not be consistent in the defects taken from the production line.

## 2.3 Types of closures

Most of the closures produced are white with a grey liner. Coloured closures are produced but not in the same volume. As a result the problem of finding defects on the white closures is to be addressed first.

There are two different types of closures produced. At first glance they look similar as the physical dimensions as well as the appearance of the outside of the closures are the same. Closer inspection reveals a number of differences. The one type of closure is used on glass bottles and the other on the polyethylene (plastic) bottles. To determine which closure is being referred to, the closures for glass bottles will be referred to as *glass* closures and the closures for the polyethylene bottles as *pet* closures.

This section gives a description of the differences between the closures as well as the effects these differences have on taking images of the closures. The basic differences are summarised in Table 2.1.

Figure 2.1(a) and (b) show typical examples of a glass and a pet closure. Certain features which are described in the following paragraphs are labelled on each image. The first difference is due to a difference in the materials used in the manufacturing process. The liner in the glass closures are always dark grey while the liner of pet closures may vary from dark grey to a very light grey, this colour difference is due to the material used for the liner. The PVC, which is a darker colour, is banned for use in food packaging in Europe and America. As a result closures produced for export use a non-PVC plastic liner, this is a lighter grey. Defects are more common in closures lined with a non-PVC plastic due to different physical properties. The lighter grey material used to line the closures reduces the contrast between the white closure and the grey liner. The higher the contrast between the liner and the closure in the image, the easier it is to segment the liner from the closure for inspection.

The second significant difference between the two closures is the way in which the edge of the liner meets the closure. On the glass closures the liner joins the closure almost perpendicularly, angle $\theta$ in Figure 2.2(a), as opposed to the pet closure where the liner joins at about 45°, angle $\phi$ in Figure 2.2(b). The effect of this is that in the image the boundary between the liner and the closure is less distinct in the case of the pet closure. In the case of the glass closure there is the benefit of the shading from the angle at which the liner joins the closure. In the case of the pet closure, the angle of the edge of the liner to the camera and the fact that the liner is fairly reflective means the liner appears even lighter in the image. This causes the boundary between the liner and the closure to be less distinct.

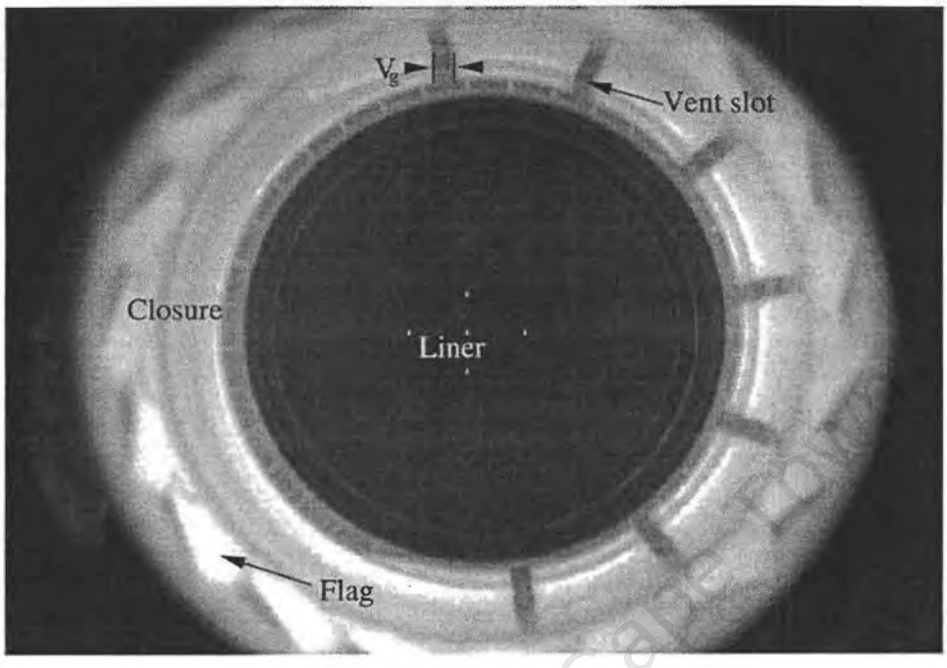
The third difference is that the liner on the pet closure is thinner than that of the glass closure. This can been seen from the measurements $p$ and $g$ in Figures 2.2(a) and (b). The thinner liner on the pet closures may cause the liner to appear lighter as the white of the closure is visible through the liner because the lighter grey plastic is slightly translucent.

The fourth difference is a difference in the construction of the closure. Figures 2.1(a) and 2.2(a) show the vent slots in the glass closures, $v_g$, are narrower than the vent slots in the pet closures, $v_p$, shown in Figures 2.1(b) and 2.2(b). This influences the lighting of the closure over the half of the closure which has vent slots. The vent slots receive less light and hence appear darker than the rest of the closure in the image. A second effect of the wider vent slots in the pet closure is that there is reduced reflection of light from the inside of the closure and this affects the uniformity of the light.

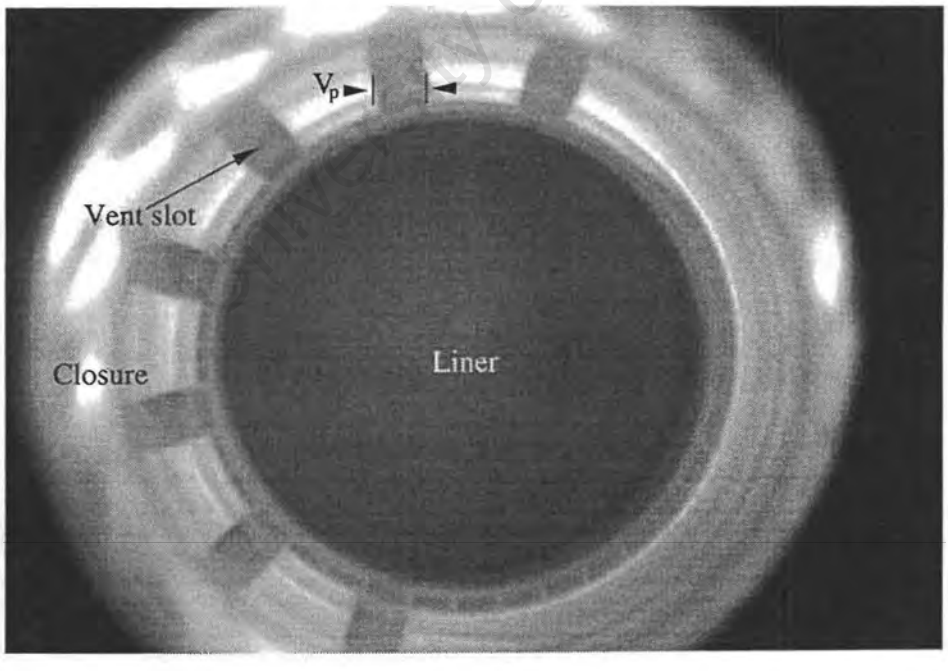
### 2.3.1 Physical dimensions of the closures

This section gives a brief description of the physical size of the closure. These dimensions are used to determine the resolution of the image. If the resolution is known, it is possible

Section 2.3: Types of closures

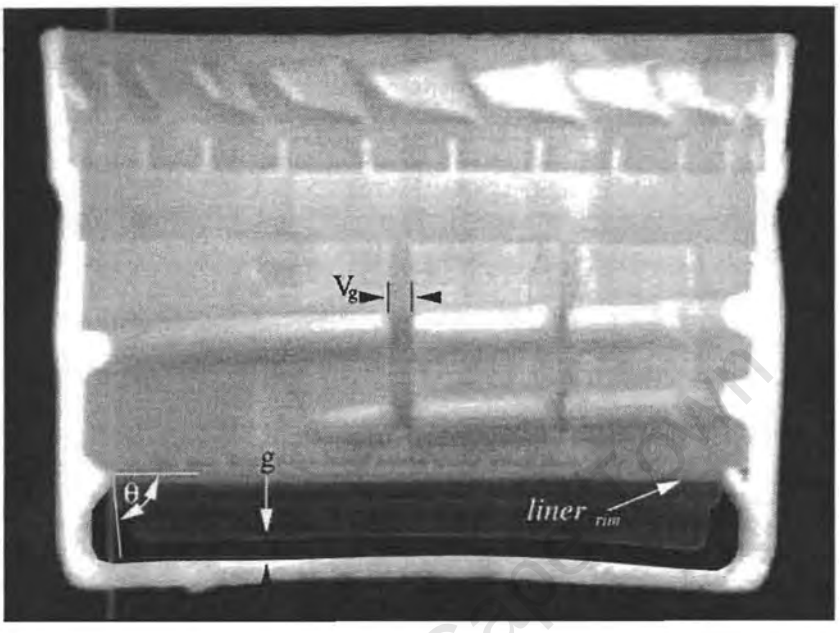



(a) Glass closure.

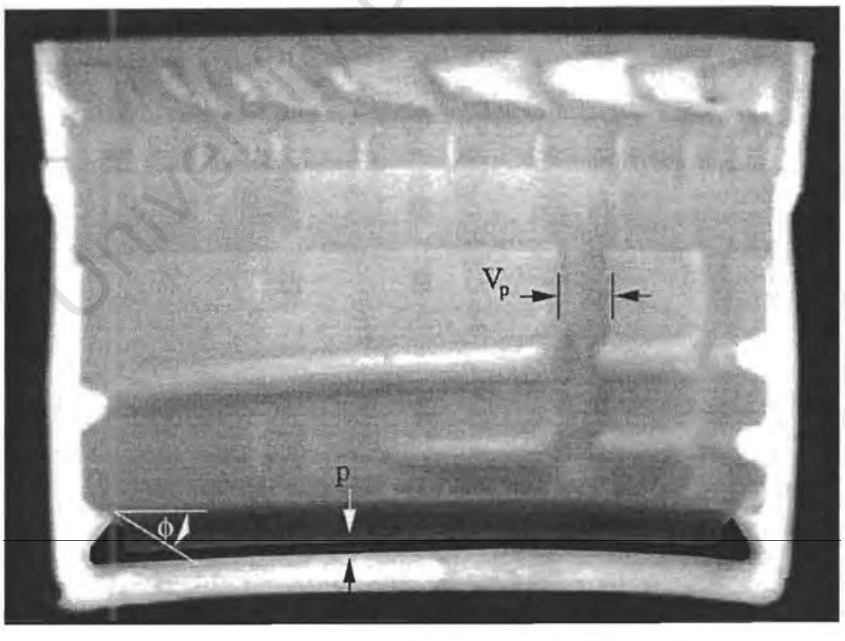



(b) Pet closure.

Figure 2.1: Examples of the two types of closures viewed by the camera in the inspection system.




(a) Glass closure.




(b) Pet closure.

Figure 2.2: Cross sections through a glass and a pet closure highlighting the important features.

| Glass Closure | Pet Closure |
|---|---|
| Liner generally dark grey | Liner can vary from dark to light grey |
| Liner meets the closure at about 90° | Liner meets the closure at about 45° |
| Thick liner($g$) | Thin liner($p$) |
| Narrow vent slots($v_g$) | Wide vent slots($v_p$) |

Table 2.1: Summary of differences between glass and pet closures.

to determine the physical dimensions of the defect. The closure dimensions are shown in Figure 2.3.

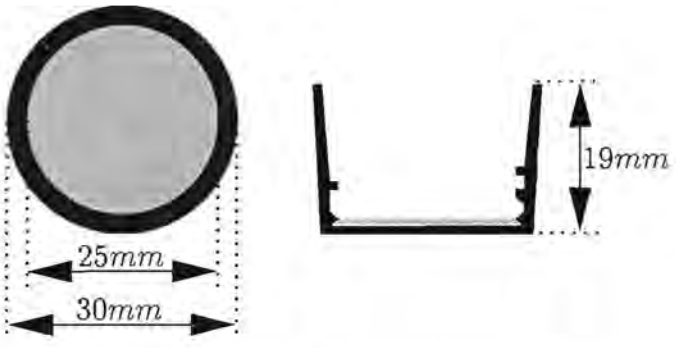



Figure 2.3: Dimensions of a plastic bottle closure.

## 2.4 Types of defects

There are a number of defects that can occur in the manufacturing process. These defects include flashing, non-fills, impurities in the liner, broken flags on the closure or a badly moulded closure. Examples of some of these defects are shown in Figures 2.4(a) and (b) and Figures 2.5(a) and (b). Another possible defect is related to the scoring of the ring of flags. When the lid is unscrewed, this ring breaks off from the rest of the closure indicating the seal on the bottle has been broken. If the scoring is too deep the ring may break off when sealing the bottle. If the scoring is not deep enough then the ring will not break off when the closure is unscrewed. The two most common defects are flashing and non-fills.

Flashing occurs when the lining material over-fills the region in the base of the of the closure which is designed to contain it. An example of this defect can be seen in Figure 2.4(a). This flashing may be due to a number of reasons. If the pellet of lining material does not land in the centre of the closure before the liner is formed it may be forced over the side, when this

happens a non-fill may occur on the diametrically opposite side of the closure. This is not very common. If the pellet of lining material is too large then some of the material will be forced over the edge of the area in the closure that should contain it. Flashing generally occurs due to mechanical imperfections in the lining machine. The size of the flashing can extend from very small to large. A tiny flash may be acceptable but larger flashes are considered to be defects.

The effect of the flashing is to cause an imperfect seal when it is placed on a bottle. This allows the liquid and the gas to escape. The plastic of the flash may also break off into the bottle during bottling. This defect is common on both the pet and glass closures.

Non-fills may be considered to be the opposite of flashing. If the liner material does not completely fill the area it is supposed to occupy, a non-fill is formed. An example of this defect is shown in Figure 2.4(b). This defect is caused if the pellet of lining material is too small. Alternatively if the small air escape channels on the tool which forms the liner become blocked an air pocket is formed which prevents the liner material from spreading correctly. If the non-fill formed is hidden under the liner rim, ($liner_{rim}$) in Figure 2.2(a), it will be invisible to the camera. The size of a non-fill may vary from almost insignificant in size to approximately 5 $mm$ in length along the edge of the liner.

If a closure with a non-fill is placed on a bottle it fails to seal correctly. This causes the bottle to leak liquid or gas. This defect is common on the glass closures but it is rare on the pet closures.

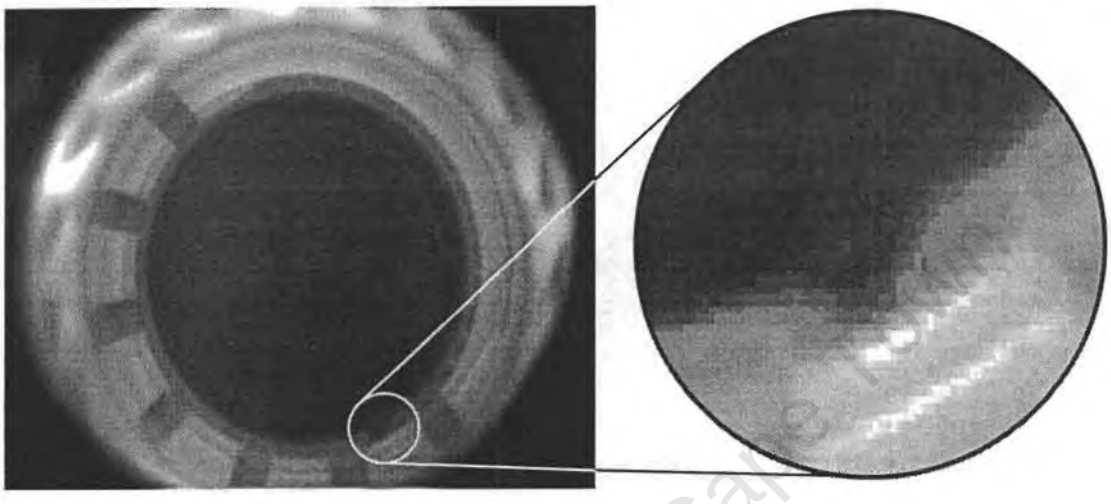
The other defects shown in Figures 2.5(a) and (b) are not very common and finding these defects is beyond the scope of this thesis.



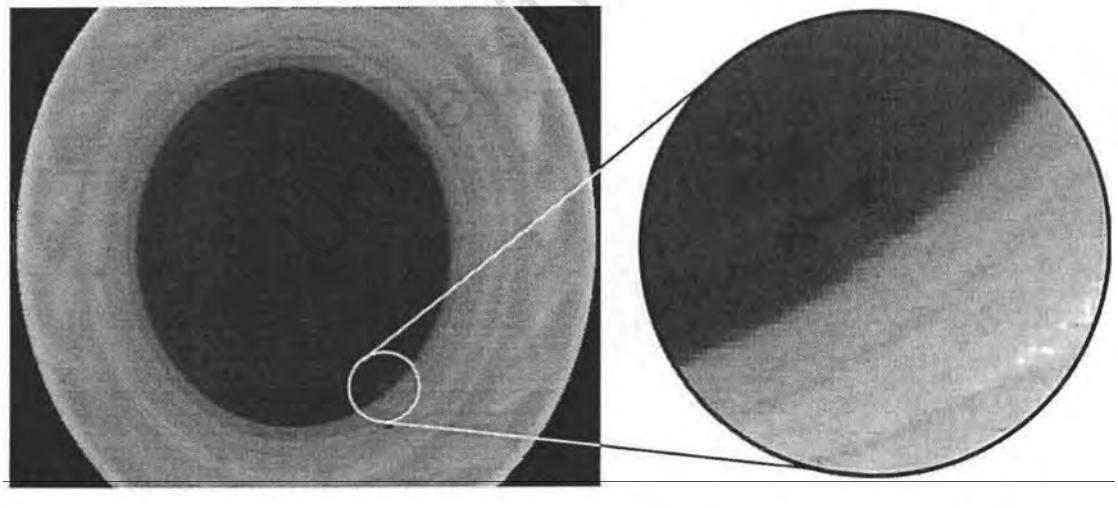
## 2.5 Specification of the inspection results

The ideal result of the inspection is to reject every defective closure while retaining every good closure. A solution must achieve results as close to this ideal result as possible. This section presents the desired results from the manufacturers point of view and the results which could be achieved in this project.

The ideal situation for the manufacturers is to have every defective closure removed from the production system while ejecting very few good closures. This situation is not possible as there is a grey area where closures with tiny defects should be classified as good. A point needs to be specified where the cutoff between good and defective closures can be set. It is not possible to detect every defect with absolute certainty, particularly within a limited period

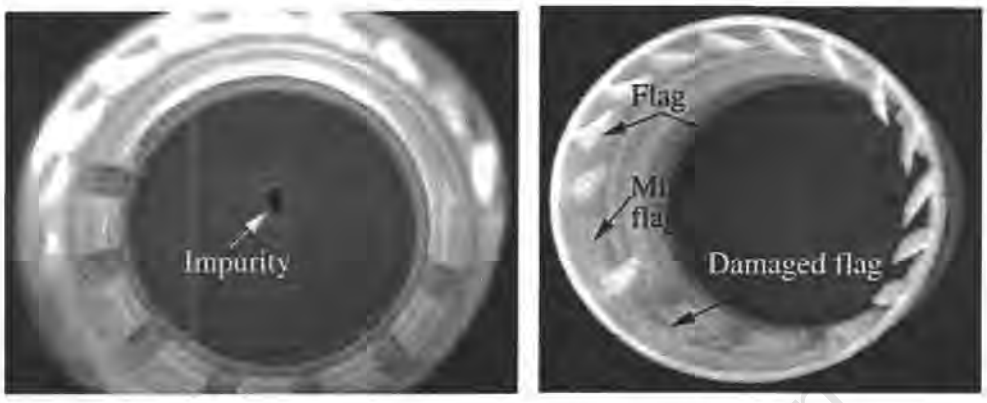


(a) Flashing.



(b) Non-fill.

Figure 2.4: Images showing the two main defects, non-fills and flashing.




(a) PVC impurities. (b) Broken flags.

Figure 2.5: Two possible closure defects which will not be examined in this thesis.

of time. There will also be a degree of misclassification with good closures being mistakenly classified as defective and vice versa.

The proposed system must be capable of ejecting the maximum possible percentage of defective closures to ensure a high quality product while on the other hand a minimum number of good closures must be ejected to ensure that waste is minimised. Taking the points above and the needs of the manufacturer into consideration, the proposed system must be capable of ejecting at least 98% of non-fills and flashing defects while not ejecting more than 0.05% of the good closures.

## 2.6 An automatic machine vision system

Taking the problem and the desired solution into consideration as well as factors such as cost, an automatic machine vision inspection system should be designed. It must be capable of inspecting every liner for non-fill and flashing defects. This must be done at 20 inspections per second. The result of the inspection should achieve the results specified in Section 2.5.

To begin the development of the automatic machine inspection system, some theory needs to be examined. This is done in the following two chapters. Chapter 3 presents the optical theory which is important in order to get good quality images for processing. Chapter 4 presents image processing theory used to develop algorithms to detect the closure defects.

# Chapter 3

# Optical Theory and Implementation



## 3.1 Introduction

This chapter gives an overview of the theory and the implementation of the optical components of the system. This involves presenting some basic definitions of optical terms in Section 3.2. A discussion of blur, Section 3.3, due to the motion of the closures relative to the camera. The effect of the lens on the image, Section 3.4. Section 3.5 discusses depth of field. In Section 3.6 the combined effects of the optical components are discussed. The values for the shutter speed and aperture setting are given.

## 3.2 Basic optical terms

Some basic terms and geometrical relationships that are used in the following sections are introduced [21, pages 209-211].

**Spatial frequency** is the number of cycles per metre.

**Modulation Transfer Function(MTF)** is the sine-wave spatial frequency amplitude response of a device in an optical system, it is one for low frequencies. An overall MTF can be found by cascading the MTFs of the separate components of the system together and finding the product.

**Thin lens** is a converging or diverging lens which has negligible thickness in comparison to the object and image distance.

**Focal length** , $f$, is the image distance when the object is infinitely distant.

**Thin lens formula** describes the relationship between the focal length, $f$, the distance to the object, $o$, and the distance to the image,$i$, shown in Figure 3.5. It is given by

$$\frac{1}{f} = \frac{1}{o} + \frac{1}{i} \tag{3.1}$$

**Image magnification** $m$ is defined as

$$m = \frac{i}{o} \tag{3.2}$$

**F**$/_{number}$ is a measure of the angular acceptance of the lens

$$F/_{number} = \frac{f}{D} \tag{3.3}$$

where

$D$ is the entrance pupil size or aperture of the lens (assuming the lens has a circular cross section)



## 3.3 Motion blur

Motion blur or image smear in an image is due to the object moving with respect to the camera while the shutter is open. There will always be some blur when taking an image of a moving object. This blur can be minimised if the shutter speed is fast, in other words having a short exposure time. This ensures that there is only a small movement of the object while the shutter is open. A blurred image will affect any further processing of the image and it is important to examine its causes and how to minimise it.

Blur can be considered in one of two domains, either the space domain or the spatial frequency domain. The blur is first considered in terms of spatial frequency and the MTF and this is then related to the space domain which gives a physical view of its cause.

The modulation transfer function for image blur on the image plane is [18, page 58].

$$MTF_{blur}(\mu_i) = \frac{\sin\left(\frac{\mu_i \gamma m}{4\pi}\right)}{\frac{\mu_i \gamma m}{4\pi}} \tag{3.4}$$

where

$\mu_i$ is the spatial frequency in cycles per metre on the image plane

$\gamma$ is the distance the object moved on the object plane during exposure

$m$ is the magnification

The distance moved by the object during exposure is

$$\gamma = Tv \tag{3.5}$$

where

$T$ is the period of exposure

$v$ is the speed of the object in the image

### 3.3.1 Movement of the closures

This section gives the implementation in terms of the system. The speed at which the closures pass the camera can be determined from examining the star-wheel shown in Figure 3.1. There are ten positions for closures on the star-wheel with $\phi = 36°$. The radius of the star-wheel is $R = 90\ mm$, this is sufficiently large to consider the movement of the closure to be linear, in a tangential direction, as it passes the camera. There are 10 closures/revolution and from Table B.1 the time for one revolution is $10 \times 0.05621\ s$ or $0.5621\ s$/revolution. The approximate radius to the centre of a closure in the star-wheel is $80\ mm$. The circumference is

$$\text{circum} = 2\pi r = 502\ mm/\text{revolution}$$

The speed of the closure is given by

$$\begin{aligned} v &= \frac{502\ mm/\text{revolution}}{0.5621\ s/\text{revolution}} \\ &= 894\ mm.s^{-1} \end{aligned} \tag{3.6}$$

The focal length of the lens used is $f = 6\ mm$. Red light is used for illuminating the closure and has a wavelength of approximately $\lambda = 650\ nm$. The object distance is 41 $mm$ and using Equation 3.1 the image distance, $i = 7\ mm$. Using Equation 3.2 the magnification $m = \frac{7\ mm}{41\ mm} = 0.17\times$. The CCD array is $512 \times 512$ pixels and is 4 $mm$ square. This gives a sampling period of,

$$\begin{aligned} T_\beta &= \frac{0.004m}{512} \\ &= 7.81^{-6}\ m \end{aligned} \tag{3.7}$$

and the sample frequency is $\frac{1}{T_\beta} = 128$ cycles/$mm$.

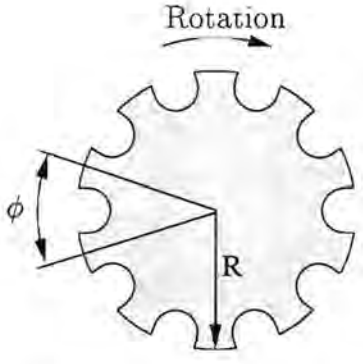



Figure 3.1: Star-wheel.

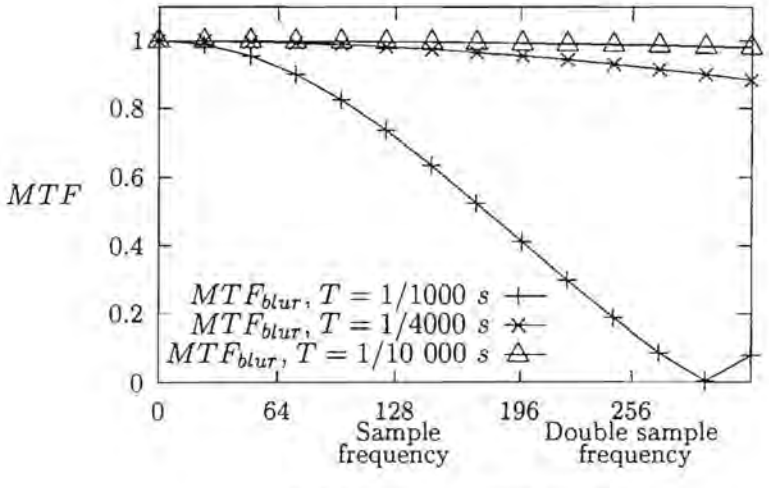
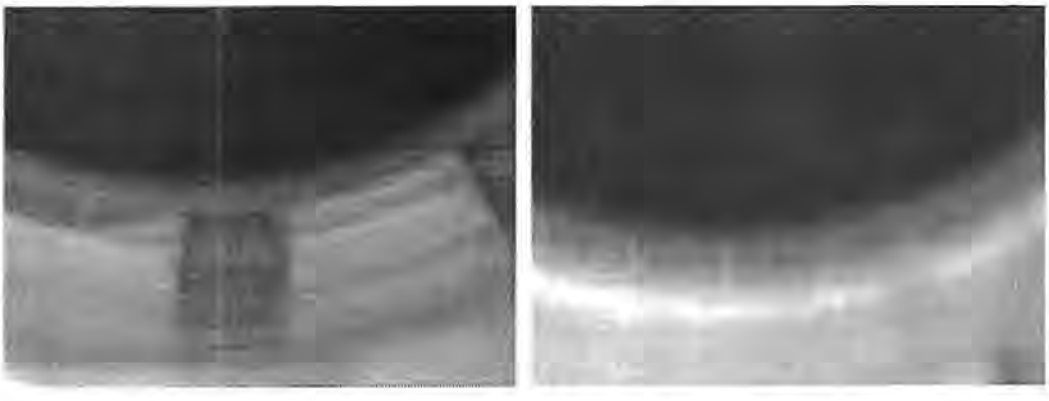



Figure 3.2: The MTF for image blur expected for three different shutter speeds.

Figure 3.2 shows $MTF_{blur}$ for three shutter speeds: $\frac{1}{1000}s$, $\frac{1}{4000}s$ and $\frac{1}{10\,000}s$. It is clear the that faster the shutter speed, the less the attenuation of the higher frequency image components. Note that the attenuation at the sample frequency is small for the two higher shutter speeds. However, if the shutter speed is slow, $\frac{1}{1000}s$, the attenuation of image components at the sample frequency is about 30%. Figure 3.3(a) shows a section of a closure, taken at $\frac{1}{10\,000}s$, which is quite crisp. Figure 3.3(b) shows a closure when the shutter speed is too slow, $\frac{1}{1000}s$. There is a definite blurring of the boundary between the liner and the closure. Other features that are visible in the first image are no-longer visible, note the moulding marks visible on the liner in the first image.



(a) Shutter speed $\frac{1}{10\,000}s$ (b) Shutter speed $\frac{1}{1000}s$

Figure 3.3: Sections of two closures taken at different shutter speeds show the effect of motion blur.

The blur can also be examined in terms of over how many pixels the closure moves while the shutter is open. In Figure 2.3 the liner is 25 $mm$ in diameter, this corresponds to approximately 180 pixels, thus the physical dimension represented by a pixel, is $\frac{25\ mm}{180} = 0.139\ mm$/pixel on the object plane. From Equation 3.6, the speed of the closures is $v = 894\ mm.s^{-1}$. The following table shows the number of pixels the closure moves during different exposure times. This comparison shows how the attenuation of the high frequency image components in the image correspond to the movement of the closure by a number of pixels.

| Shutter speed | Distance moved | Number of pixels moved |
|---|---|---|
| $\frac{1}{10\,000}s$ | 0.089 mm | 0.64 |
| $\frac{1}{4000}s$ | 0.223 mm | 1.61 |
| $\frac{1}{1000}s$ | 0.894 mm | 6.43 |

A movement of about 1.6 pixels is tolerable, this is where the $MTF_{blur}$ is about 0.97 at the sample frequency.

## 3.4 The lens

The $MTF$ of the lens on the image plane is [21, page 213]

$$MTF_{lens}(\mu_i) = \frac{2}{\pi}\left[\arccos(\lambda F \mu_i) - \lambda F \mu_i \sqrt{1-(\lambda F \mu_i)^2}\right] \tag{3.8}$$

and has a spatial cutoff frequency of $\mu_i = \frac{1}{\lambda F}$cycles/metre.

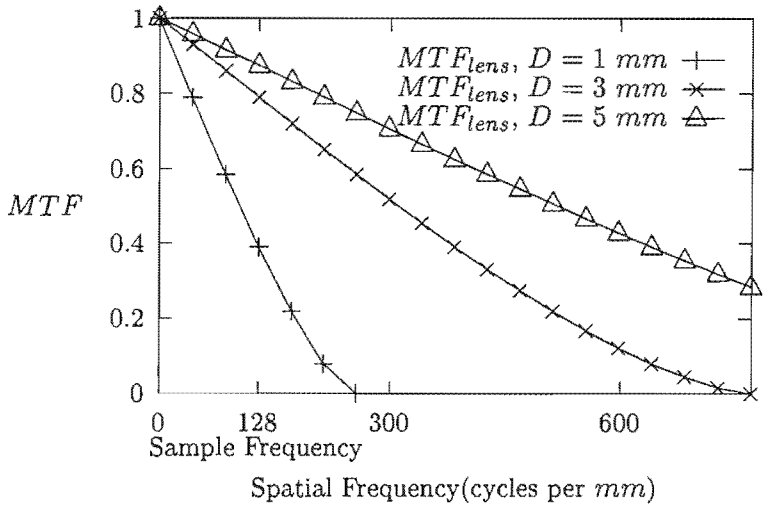



Figure 3.4: The $MTF_{lens}$ shown for the aperture varying from 1 to 5 $mm$.

From Figure 3.4 shows the $MTF_{lens}$ for the aperture varying $D$ from 1 to 5 $mm$. This has the effect of varying the $F/_{number}$ of the lens. This shows that the smaller the aperture, the greater the effect of the lens attenuating the higher frequency components of the signal.

## 3.5 Depth of field

An image can only be perfectly focused at a single distance from the lens. An image of a flat surface cannot be in focus across the whole image as all points are not the same distance from the lens. The depth of field is defined as the distance over which the circles of confusion remain smaller than the resolution of the imaging device.

Points at a distance $i$ from the lens are imaged as points. Points at other distances from the lens on the image, $i'$, are imaged as circles, known as circles of confusion, which have radius(Falk [8])

$$\frac{d}{i'}\left|i' - i\right|$$

where $d$ is the diameter of the lens.

In Figure 3.5, consider a point at the centre of the image. The angular size of the circle of confusion is represented by $\phi$. The image plane is a distance $i$ from the lens. The size of a single pixel on the image plane, $T_\beta$ specifies the resolution of the imaging device. This is the same as the sampling interval, $T_\beta = 7.81 \times 10^{-6}\ m$, Equation 3.7. The angle $\phi$ is given by

$$\begin{aligned} \frac{\phi}{2} &= \arctan\left(\frac{\frac{T_\beta}{2}}{i}\right) \\ \phi &= 2\arctan\left(\frac{T_\beta}{2i}\right) \end{aligned} \tag{3.9}$$

As $i$ is very large relative to $\frac{T_\beta}{2}$ then

$$\phi = \frac{T_\beta}{2i}$$

The distance of the point of focus from the lens is $o$. The near limit of the depth of field, $D_{near}$, and the far limit of the depth of field ,$D_{far}$, are a measure of how far the object can move towards or away the lens before it is unfocused. $D_{near}$ and $D_{far}$ are defined by Morgan [19, pages 42-45]

$$D_{near} = \frac{o^2\tan(\phi)}{d + o\tan(\phi)} \tag{3.10}$$

$$D_{far} = \frac{o^2\tan(\phi)}{d - o\tan(\phi)} \tag{3.11}$$

The aperture diameter has the largest effect on the depth of field. To have a large depth of field, the aperture needs to be small and the depth of field increases with increasing object distance.

If a lens with a focal length of $f = 6\ mm$ and the object distance of $o = 41\ mm$ is used then Equations 3.1 and 3.9 give $\phi = 0.00111\ rad = 0.06°$. The plots of $D_{near}$ and $D_{far}$ for these values are shown in Figure 3.6. In Figure 3.7(a) a comparison of the depth of field as the aperture changes from 1 $mm$ to 5 $mm$ for object distances of 30, 40 and 50 $mm$. This shows that the depth of field increases as the object is moved further away. Figure 3.7(b) shows a comparison of the near and far limits of the depth of field vary as the focal length, $f$, is varied from 4 to 8 $mm$. This indicates that a lens with a shorter focal length will have a greater

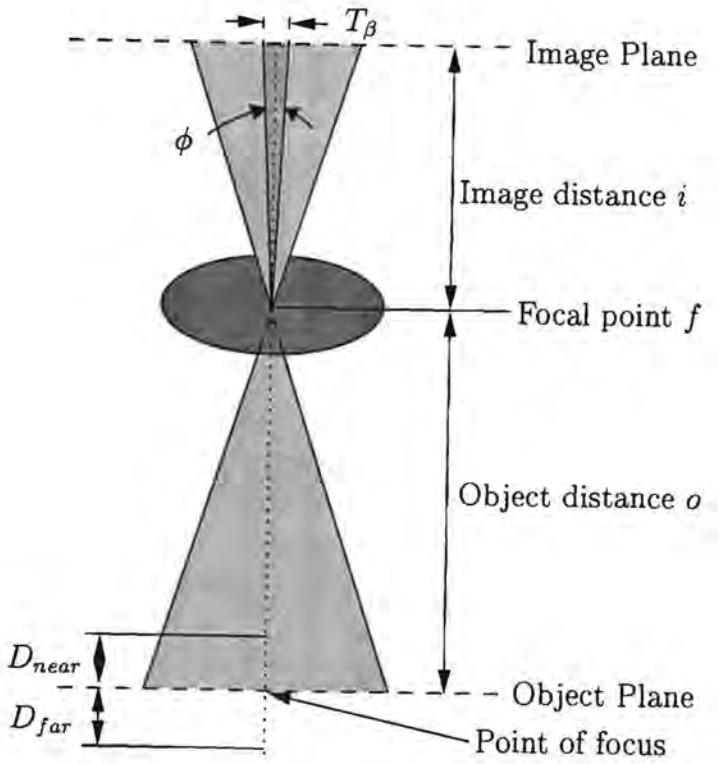



Figure 3.5: Depth of field related to the near field and far field.

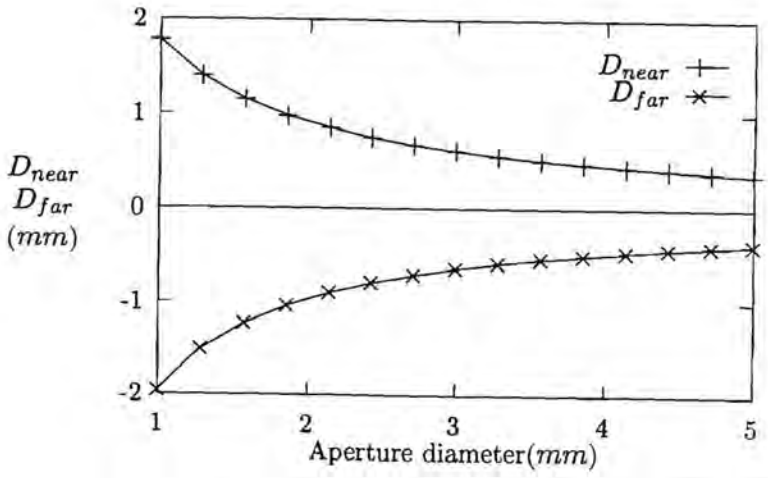



Figure 3.6: The near and far limits of the depth of field are shown as a function of the aperture diameter with the zero line being the point of focus.

depth of field.

## 3.6 Combined effects of the optical components

This section examines the combined effects that are examined in Sections 3.3, 3.4 and 3.5 covering motion blur, the lens and depth of field. The overall $MTF_{sys}$ is determined. This is done ignoring the $MTF$ of the detector in this system as it is not a significant source of attenuation(Moon [18]) until about three times the sample frequency and as a result it is not included. The overall $MTF$ of the system is simply the product of the individual $MTFs$.

Figure 3.8(a) shows the combined

$$MTF_{sys} = MTF_{lens} \times MTF_{blur}$$

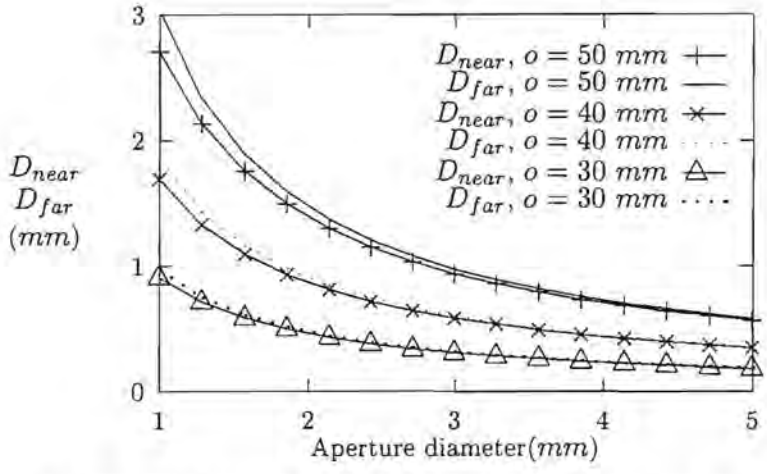
for a shutter speed set at $T = \frac{1}{1000}s$ and an aperture of $D = 1\ mm$. This shows the effect of multiplying the two separate $MTF$s together.

Table 3.1 shows the trends for the overall $MTF_{sys}$ and the depth of field when varying the shutter speed $T$ and the aperture diameter. From the table it is clear that having a fast shutter speed and a wide open aperture will give a good $MTF_{sys}$. The wide aperture however, gives poor depth of field which is not desirable.

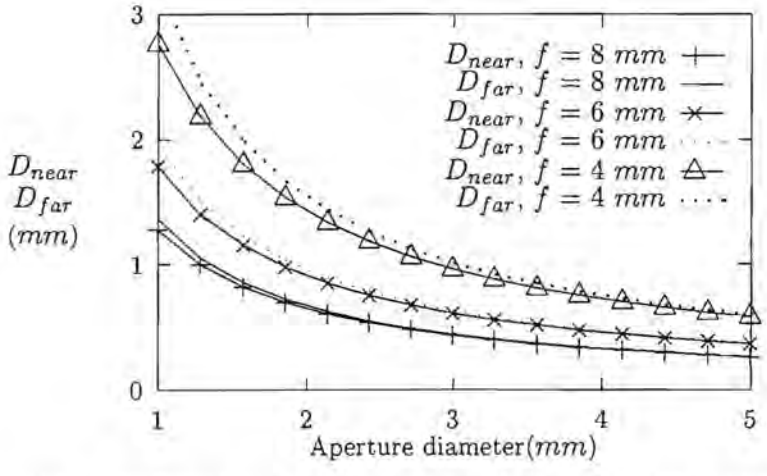
| Aperture size $D$ in $mm$ | Shutter Speed $T$ in $s$ | $MTF$ features | Depth of field |
|---|---|---|---|
| small(1 $mm$) | medium$\left(\frac{1}{4000}s\right)$ | $MTF_{lens}$ very dominant | good |
| large(5 $mm$) | medium$\left(\frac{1}{4000}s\right)$ | good overall $MTF_{sys}$ | poor |
| medium(3 $mm$) | slow$\left(\frac{1}{1000}s\right)$ | $MTF_{blur}$ very dominant | fair |
| medium(3 $mm$) | fast$\left(\frac{1}{10\ 000}s\right)$ | $MTF_{lens}$ dominant | fair |

Table 3.1: Trends of the overall $MTF$.

Many factors need to be taken into consideration when choosing a lens, a shutter speed and the aperture setting which are dependent on the lighting available. Based on the factors discussed in the previous sections and the available equipment, Figure 3.8(b) shows the combined $MTF_{sys}$ for this system. $T = \frac{1}{4000}s$ and $D = 3\ mm$ giving a reasonable $MTF$ where the

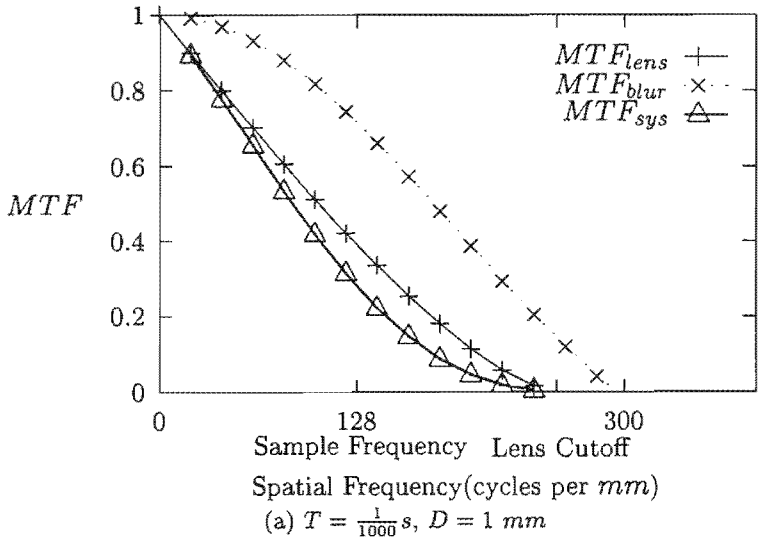



(a) $D_{near}$ and $D_{far}$ as the object distances vary with the focal length fixed at 6 $mm$.

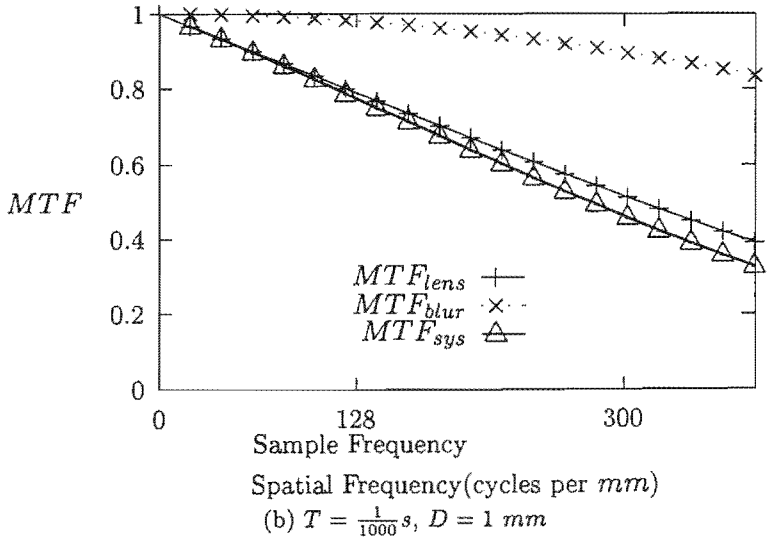



(b) $D_{near}$ and $D_{far}$ as the focal length varies with a fixed object distance of 41 $mm$.

Figure 3.7: Showing the near and far limits for the depth of field as the object distance varies in (a) and as the focal length varies in (b).




(a) $T = \frac{1}{1000}s$, $D = 1\ mm$




(b) $T = \frac{1}{1000}s$, $D = 1\ mm$

Figure 3.8: Combined $MTF_{sys} = MTF_{lens} \times MTF_{blur}$

depth of field is acceptable.

The optical theory and implementation discussed here allow a good image to be taken. This image must be processed in order to find defects in the closure in the image. The theory that is applicable to the design of algorithms to detect the defects is given in the next chapter.

# Chapter 4

# Image Processing Theory



## 4.1 Introduction

In this chapter basic image processing theory that is relevant to the work done in this thesis is presented. This includes edge detection, Section 4.2 and thresholding, Section 4.3. These are simple operations which can be applied either locally, operating on pixels in a small neighbourhood or to an entire image.

The rest of the chapter discusses morphological image processing for binary images, Section 4.5, and grey-scale images, Section 4.6. Morphological image processing involves using small shapes to probe an image to determine its geometrical structure.

## 4.2 Edge detection

Edge detection is the process by which edges are found in an image. Horn [14, page 161] describes edges as curves in an image where rapid changes in the brightness or in the spatial derivatives of the brightness occur. Hussain [15, page 102] describes a simple edge as a sharp discontinuity in grey-level profile in an image. Haralick and Shapiro [12, page 337] state that an edge is the boundary between two pixels when their brightness is significantly different. Edges in an image give information about the objects in the image. Among other things, edges show where the reflectance or colour of the object changes, the presence of shadows and where the orientation of the surface changes abruptly. If these edges can be found they give information about the features of the object which they represent. This means edge information provides a method of segmenting images into different regions.

### 4.2.1 Performance of edge detectors

According to Canny [2] when designing an optimal edge detector, three performance criteria need to be considered.

**Good detection** The probability of not marking an edge point as an edge and of marking a non-edge point as an edge need to be as low as possible. This is done by maximising the signal to noise ratio.

**Good localisation** The points that are marked as an edge point should be as near as possible to the centre of the true edge.

**Single response** The edge detector should respond only once to each edge.

Noise in the brightness of the image limit how accurately the edges in an image can be found. According to Horn [14, page 170] there is a trade-off between sensitivity to edge information and accuracy. Many edge detectors are simple differential operators which amplify high frequency noise. As a result edges with a low contrast are not distinguishable from noise in the output. An operator with a larger region of support reduces the contribution from noise to the output as the signal difference between neighbouring regions is maintained. The difference between the average of $N$ pixels on either side of an edge $E$ in Figure 4.1 gives the difference in brightness levels. Horn [14, page 170] states that if the standard deviation, $\sigma$, is the same, for all points, then the standard deviation of the result will be $\sigma/\sqrt{N/2}$. Thus $\sigma$ can be reduced by increasing the support region of the edge operator in the horizontal and vertical directions. As the region of support becomes bigger, the chances that it will overlap other edges increases. If the region of support includes other edges then the result will be incorrect.



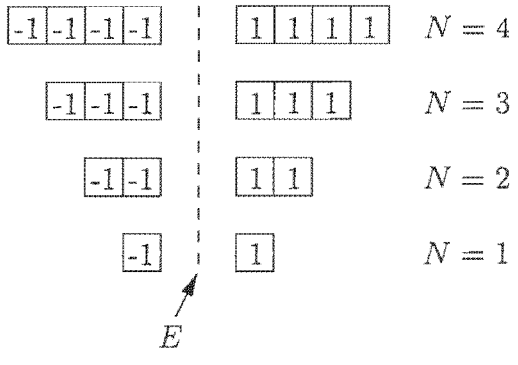


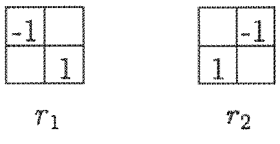

Figure 4.1: A vertical edge operator subtracting the sum of the grey-level of $N$ pixels either side of an edge.

In an image which has had the edges enhanced by some operation, the edges will appear as points with a high magnitude. Points above a certain magnitude must be chosen as edges and any points below this threshold are not edges. If the threshold is too high then weak edges may be missed. On the other hand, making the threshold too low will result in many non-edges being identified as edges. Increasing the region of support of the edge detector reduces the frequency at which changes can be identified. This reduces the effect of noise and makes weak edges more difficult to detect. This means that short edges will not be detected unless they have a high contrast.

### 4.2.2 Gradient edge detectors

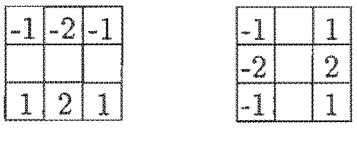
A gradient edge detector is an operation which measures a value related to the difference in brightness or the gradient of an edge as well as the direction of the edge. Haralick and Shapiro [12, pages 337-338] describe early gradient edge operators used by Roberts(1965), Sobel(1970) and Kirsch(1970). Roberts used two $2 \times 2$ masks, Figure 4.2(a), to determine the gradient in the two diagonal directions. The edge magnitude is $\sqrt{{r_1}^2 + {r_2}^2}$. Sobel used two $3 \times 3$ masks Figure 4.2(b). The gradient magnitude is $\sqrt{{s_1}^2 + {s_2}^2}$ and the gradient direction is $\arctan\left(\frac{s_1}{s_2}\right)$. Kirsch used 8 different masks to determine the edge strength in 8 directions. These gradient edge detectors are similar but vary with respect to the accuracy of the magnitude and direction of the gradient found. These operators are local operators and only operate on pixels in a small neighbourhood.






$r_1$ $r_2$

(a) Masks for the Roberts edge operator.




$s_1$ $s_2$

(b) Masks for the Sobel edge operator.

Figure 4.2: The masks used for the Roberts and Sobel edge operators.

### 4.2.3 Directional edge detection

If the direction of the edge is known beforehand then an edge detector can be orientated to find edges in that direction. This can be done if there is prior knowledge about the object and its orientation in the image. An example of an edge detector which is orientated to find edges in a vertical direction can be seen in Figure 4.1. Similar edge detectors can be made to detect edges in the vertical and diagonal directions.

## 4.3 Thresholding

Thresholding is a labelling operation on a grey-scale image [12, page 14]. In the image pixels which have grey-scale values higher than a certain level are marked as 1 and pixels which are below the set value are marked as 0, the result is a binary image. The selection of the threshold level depends on what feature or aspect of the image is to be extracted.

Thresholding can be used to separate a light object from a dark background. For thresholding to be used in a system, an automatic determination of the threshold level needed to separate the light pixels from the dark pixels. If the distribution of dark and light pixels in the image are widely separated then the histogram of the pixel intensities in the image will be bimodal and the threshold level can be set at the centre of the two modes. This method relies on the histogram of pixel intensities being known. A second method is to sample the dark pixels and the light pixels and set the threshold between these levels.

If the threshold is too high then light pixels will be incorrectly labelled as dark. On the other hand if the threshold is too low then dark pixels will be incorrectly labelled as light.

Using a single global threshold level is a very simple method that only labels single pixels as above or below the threshold level across the entire image. This is a very simplistic approach and does not take gradients in pixel intensities across the image into consideration. A global threshold can only be used under uniform lighting conditions.

## 4.4 Morphological image processing

Mathematical morphology is a branch of nonlinear image processing and analysis that concentrates on the geometric structure within an image [5, page 1]. The scope of morphological image processing is very wide, it has been used in texture analysis [25], segmentation [5], restoration [16], pattern inspection [4], feature extraction [16], [3], shape analysis [13] and edge detection [22].

Morphology is based on probing an image with a structuring element and quantifying how well the structuring element does or does not fit an image at a particular point. The structuring element uses information in a neighbourhood, the size of the structuring element, in the image. The structuring element is used to probe the image. This means the shape and size of the structuring element will determine what information is found. The fitting of the structuring element is the primary operation in mathematical morphology. Different morphological filters are built up by applying different combinations of the basic morphological operations.

Morphology can be divided into two broad categories, binary morphology which is used on binary image and grey-scale morphology which is applied to grey-scale images. Section 4.5 discusses the basic theory of binary morphology and Section 4.6 discusses grey-scale morphology.



## 4.5 Binary morphology

This section discusses morphological operation on binary images. A pixel in a binary image is either zero or not zero. There are two basic morphological operations erosion and dilation. The binary erosion operation serves as the mathematical marker of where the structuring element fits within an image [6, page 6]. Eroding an image $A$ by a structuring element $B$ is denoted by $A \ominus B$ and defined by

$$A \ominus B = \{x : B + x \subset A\}$$

Erosion has the effect of shrinking an image.

Dilation is the dual of erosion. Dilation of an image $A$ by a structuring element $B$ is denoted by $A \oplus B$ and defined as

$$A \oplus B = [A^c \ominus (-B)]^c$$

where $A^c$ denotes the complement of $A$. Dilation has the effect of expanding an image.

Opening an image $A$ by a structuring element $B$ is denoted by $A \circ B$ and defined as

$$A \circ B = (A \ominus B) \oplus B$$

Closing an image $A$ by a structuring element $B$ is denoted by $A \bullet B$ and defined as

$$A \bullet B = (A \oplus B) \ominus B$$

### 4.5.1 Detecting boundaries

The basic binary morphological operators can be combined to produce different methods of locating the boundaries in a binary image.

Erosion of an image by a square structuring element shrinks an image and dilation by the same structuring element expands an image. There are three methods of using the original image, the eroded image and the dilated image to detect the boundaries [5, pages 32-33]. Taking an image $A$ and a square structuring element $B$ the three possible methods are shown in Figure 4.3

$(A \oplus B) - A$ will give the points just outside of the boundary.

$A - (A \ominus B)$ will give the points just inside of the boundary.

$(A \oplus B) - (A \ominus B)$ will give the points on the boundary which are either side of the actual boundary.

## 4.6 Grey-scale morphology

The basic grey-scale morphological operators have the same name as those used for binary morphology. The actual operation itself is different.

### 4.6.1 Basic grey-scale morphological operators

The two primary grey-scale morphological operators are erosion and dilation. Erosion is defined as

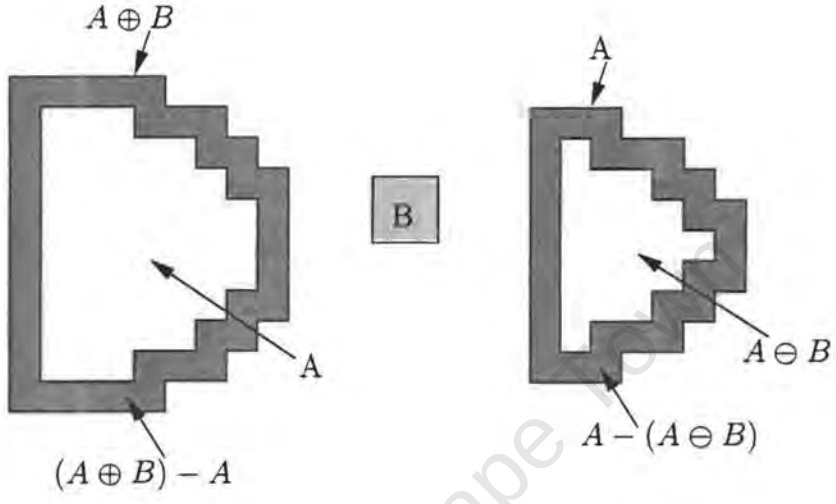
$$(f \ominus g)(x) = min\{f(z) - g_x(z) : z \in D[g_x]$$

where $f$ is the image and $g$ is the structuring element and $D[g_x]$ is the domain of the structuring element. Dilation is defined as

$$(f \oplus g)(x) = max\{f(z) + g_x(z) : z \in D[g_x]\}$$

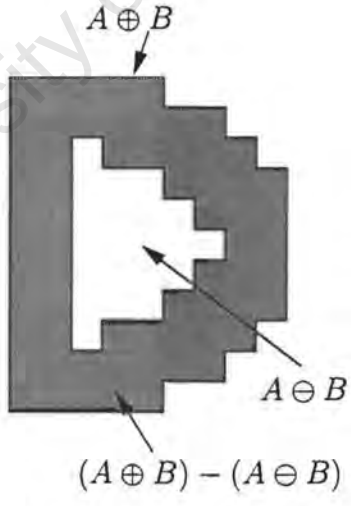
Two secondary operators are based on erosion and dilation, these are opening and closing. Opening is defined as

$$f \circ g = (f \ominus g) \oplus g$$




(a) Points outside edge.

(b) Points inside edge.




(c) Points straddling real edge.

Figure 4.3: Using binary morphological methods to find boundaries in binary images.

and closing on the other hand is defined as

$$f \bullet g = (f \oplus g) \ominus g$$

### 4.6.2 Algebraic properties

These are some algebraic properties of the grey-scale morphological operators(Dougherty [5, page 103]). Scalar multiplication by a real number is distributive to both erosion and dilation.

$$\begin{aligned} t(A \oplus B) &= tA \oplus tB \\ t(A \ominus B) &= tA \ominus tB \end{aligned} \tag{4.1}$$

Erosion is not associative. Successive erosions are equivalent to the erosion of the image by the structuring elements combined by dilation.

$$(f \ominus g) \ominus h = f \ominus (g \oplus h) \tag{4.2}$$

Dilation is associative.

$$(f \oplus g) \oplus h = f \oplus (g \oplus h) \tag{4.3}$$

Dilation is commutative.

$$A \oplus B = B \oplus A \tag{4.4}$$

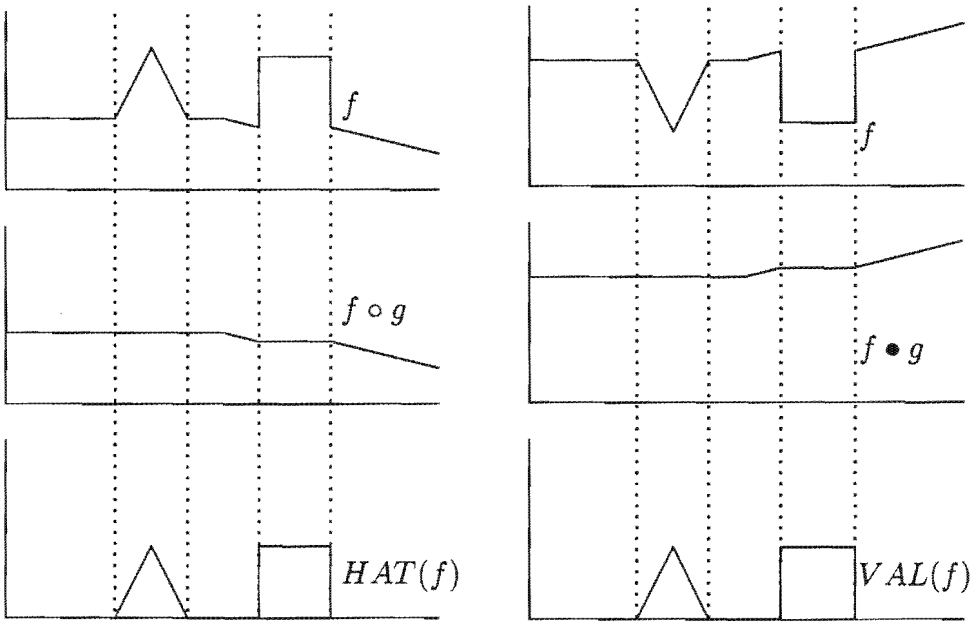
### 4.6.3 Top-hat transform operator

The operator known as the top-hat transform is

$$HAT(f) = f - (f \circ g)$$

where $f$ is the image and $g$ is an appropriate structuring element. Opening is an anti-extensive operation, this results in the opened image being below the original and $HAT(f)$ being nonnegative. Figure 4.4(a) shows a signal $f$ processed by a flat structuring element

$g$ which is wider than the peaks. The top-hat transform can be used for detecting peaks in images.




(a) Top-hat transform.

(b) Valley transform.

Figure 4.4: Illustrations of the top-hat and valley transforms.

Figure 4.4(a) also shows how a constant gradient in the right hand part of the signal $f$, which may have been caused by a gradient in the illumination of the object, has been discarded by the subtraction during the process. This transform is useful to remove gradual variation in the lighting from the image.

### 4.6.4 Valley transform operator

The valley operator is the dual to top-hat transform described in Section 4.6.3. The valley detector is given by

$$VAL(f) = (f \bullet g) - f$$

This is also a form of top-hat transform but it shall be called the valley transform. The valley transform is useful for detecting troughs in an image.

Figure 4.4 shows how this operator works, showing the signal $f$, the closed signal $f \bullet g$ and the resulting transform. Closing is an extensive operation, that is closed image lies above the

image, thus the result is always nonnegative.

### 4.6.5 Peak and valley detection

A way of determining the peaks and valleys simultaneously is to use an operator which is a combination of the top-hat and valley transforms

$$(f \bullet g) - (f \circ g)$$

The closing and the opening can be done in parallel and followed by the subtraction. The order of the operators is important as closing is an extensive operation and opening is anti-extensive and thus the resulting image is nonnegative.

### 4.6.6 Morphological gradients

The operation to find the morphological gradient is generally given by

$$GRAD(f) = (f \oplus g) - (f \ominus g) \tag{4.5}$$

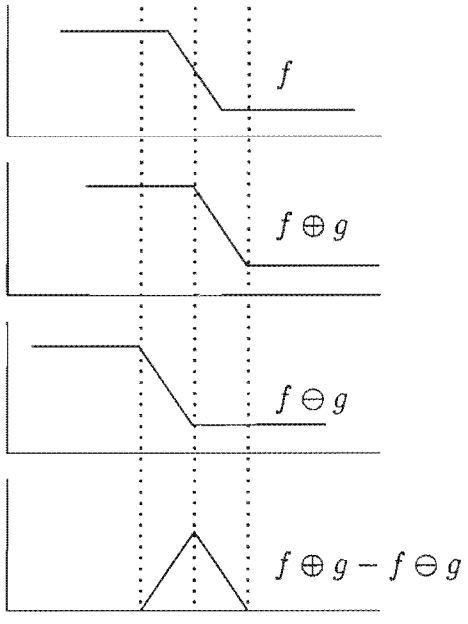
where $g$ is the structuring element. Figure 4.5(a) illustrates this method for a signal. The signal is eroded and dilated by a flat structuring element. The eroded signal is subtracted from the dilated signal giving the gradient. The edge is where the gradient is a maximum.

A second morphological gradient method, the erosion morphological gradient, is given by the operator

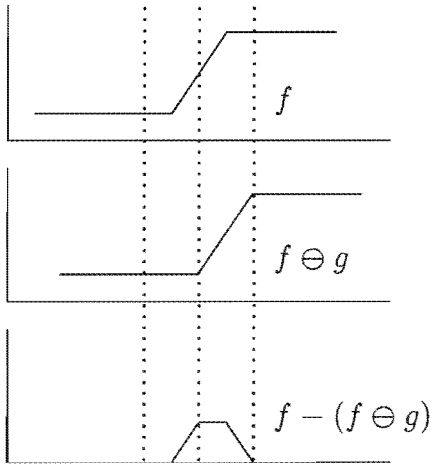
$$f - (f \ominus g) \tag{4.6}$$

where $g$ is the structuring element. This second method of determining a morphological gradient requires only a single morphological operation, an erosion. This makes the operation faster. Figure 4.5(b) shows the result of this operation on the same signal $f$, the eroded signal is subtracted from the original signal leaving a signal showing edge strength. From the figure it can be seen that Equation 4.5 gives a better result than Equation 4.6 as the second result has not got a single maximum value and the signal is smaller.

This image processing theory is the basis for the development of the algorithms used to detect defects in the closures. These algorithms are presented in Chapter 6. Before these algorithms are examined, the design and selection of the hardware is given. This is what follows in the next chapter.




(a) Morphological gradient: signal, dilation, erosion, gradient.




(b) Morphological gradient: signal, erosion, gradient.

Figure 4.5: Two methods of determining the morphological gradient. (a) from Dougherty [5, page 119].

# Chapter 5

# Hardware and Image Capture



## 5.1 Introduction

The purpose of this chapter is to describe the hardware used in the development of the machine vision system. The selection or design of the hardware used directly affects the choice and use of the image processing algorithms. This chapter describes each item of hardware used. In Chapter 7 an overview of how the different hardware components fit together within the whole system is given. Lighting is probably the most important factor in the formation of images in an image processing system. The lighting is described in Section 5.2. As this is important a lot of time went into this design. Firstly the design of an incandescent lighting system which was found to be unsuitable for use in the factory. Secondly the design of the LED lighting system which was used. Section 5.3 is about the camera. Sections 5.4, 5.5 and 5.6 describe the host computer, the DSP board and the frame grabber. The host computer is the backbone of the system while the frame grabber and DSP are used to digitise and process the image. Section 5.7 explains the development of the input/output card which receives information from the transducers and makes it available to the computer, allowing the computer to control the output devices.

## 5.2 Lighting

In machine vision applications, the lighting is generally very critical. In this application the development of a fast algorithm is important, thus the lighting should be designed in such a way as to aid and not hinder the image processing algorithm.

The image formed by the camera ,$f(x, y)$, may be looked at as the product of the reflectance

of the object, $r(x, y)$, and the illumination function, $i(x, y)$ [20]. The reflectance of the closure cannot be controlled. This means the illumination of the closure must be designed in such a way as to highlight the defects in the closure. The following factors were considered when designing the lighting system

- Illumination must be uniform around the boundary of the closure liner.
- The illumination must not cause artifacts or shadows to be visible in the image.
- Lighting should be used to highlight defects in the closure.
- No light should shine through the side of the closure as this obscures non-fills.
- The camera and lens position relative to the closure dictate the position of the lighting.
- No light should shine directly onto or be reflected onto the camera lens.
- The light intensity needs to be sufficient to allow the desired selection of shutter speed and aperture setting as specified in Section 3.6.
- The lighting system must be robust to ensure the system is reliable and low maintenance.

The camera position relative to the closure, shown in Figure 5.1, is fixed by setting up the camera and the lens to get the desired image size. The closeup image of the closure is achieved by using a lens with a short focal length. This lens distorts the closure in the image which makes it easier to see the liner boundary. The illumination of the closures needs to be accomplished without disturbing this arrangement.



### 5.2.1 Incandescent lighting system

An incandescent lighting system was chosen for the initial design which was used during the development of a prototype. This allowed even illumination in the closure of a high enough intensity. To do this a very bright diffuse light source was needed. This was achieved using four 12 volt, 50 watt halogen spot lights with a beam angle of 38°. In order to diffuse the light from the spot lights opaque perspex was placed in the beam, which was reflected off a matt white surface. To ensure even illumination, the four light sources were placed in a symmetrical pattern on the corners of a square with the camera in the centre. To prevent the light from shining through the sides of the closure, a screen was placed above the closure to ensure that the only light reaching the closure was through the hole above the closure. The layout is shown in Figure 5.2.

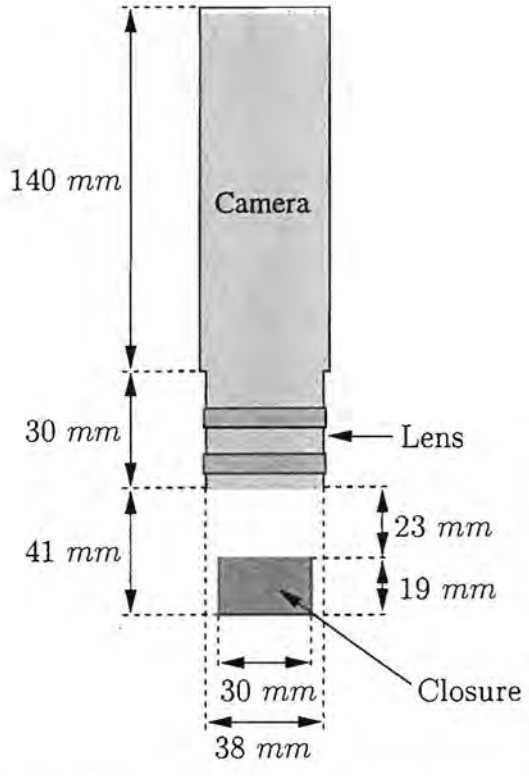



Figure 5.1: The relative position of the camera and closure are shown. Note the limited space available for illumination.

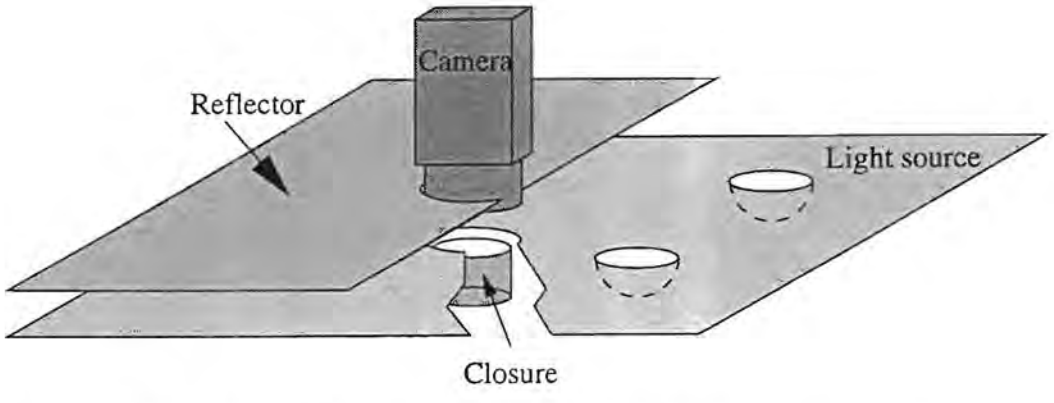



Figure 5.2: The position of the incandescent lights set up around the camera.

## Section 5.2: Lighting

Very little of the light produced by the light sources actually contributed towards illuminating the closure. Unnecessary extra heat was generated. The light sources needed cooling to prevent overheating of the camera. The heat discoloured the paint used on the reflector. The cooling was done by allowing compressed air to flow over the light sources.

The illumination of the closures using this method was very good. The closures were illuminated with uniform light of sufficiently high intensity. Defects in the closure's liner were easily visible. This lighting system was not acceptable as it was not sufficiently robust.

### Problems with the incandescent lighting design

Although the incandescent lighting design illuminated the closure well, there were problems with the system. These problems were the inefficiency and unreliability of the lighting system. The level of maintenance required was far above an acceptable level for system.

Two factors contributed to the high maintenance necessary for the lighting system. The first was that a gas escaped from the hot PVC. This condensed onto the reflectors of the bulbs making them dull. This reduces the effectiveness and made it necessary to clean them regularly, more than once a day. A second factor adding to the dulling of the light source reflectors was the compressed air used to cool the lights. Although this air was filtered it was still dirty and after running the system on the lining machine for a few days the reflective surfaces became blackened. Both these factors reduced the intensity of the lighting. It might also have been a factor in reducing the life of the bulbs.

The second factor was the life of the bulbs used. This was guaranteed to be approximately 750 hours [7] or approximately 30 days of continuous use. This added significantly to the running cost to the system. The illumination with this method was only usable if all four light sources were operational. The short life of the light bulbs reduced the overall reliability of the system.

The footprint of the lighting system was large compared to the space occupied by the camera. As space is limited on the lining machine this was found to be unsuitable.

The incandescent lighting system was suitable for use in the development of the prototyping and testing of the system. The reliability and the high maintenance requirement of the lighting setup made it unsuitable for long term use in the factory.

### 5.2.2 LED lighting system

The problems with the incandescent lighting system were not impossible to work around. However, based on the needs in the factory, it became desirable to have a lighting system which had a smaller footprint and was more robust and required less maintenance while yielding the same or better illumination of the closure.

The lighting system was redesigned with the following factors taken into consideration

- Lights must have a small footprint.
- No forced cooling must be necessary. This will prevent unnecessary dirt from being brought into the optical system.
- Lights will not be placed below the camera lens allowing more freedom in the placing of the camera.
- Lights need to have a long lifespan.

To achieving these goals, light emitting diodes(LEDs) were chosen as the light source. These have many advantages over the light bulbs used in Section 5.2.1. They have an extremely long lifespan and have a much lower power consumption as the light is focused where it is needed and there is little heat generated.

The problem with using LEDs is that they provide a focused beam which is not ideal to create a uniform illumination. As the inspection algorithm only examines the liner boundary only a ring of uniform light needs to be created. The focused beams of the LEDs can be pointed through the gap between the lens and the closure. This was done using a ring of 24 LEDs placed around the lens. The focused beams were defused by frosted glass which is clear in the centre at the lens. Figure 5.3 shows the arrangement of the LEDs around the lens. The height of the camera relative to the closure is fixed, this fixes the angle of the LEDs, necessary to illuminate the liner boundary. The angle of the LEDs is fixed by the geometry of the system. The base of the lens is 41 $mm$ from liner and horizontal distance is 44 $mm$ thus the angle of the LEDs is approximately 43°.

#### Observations regarding the use of the LED lighting system

The light produced by the LED lighting system is not as uniform as that produced by the incandescent lighting system. As a result the image produced is not as good. However in the part of the image that the algorithm examines the light is uniform and the image in

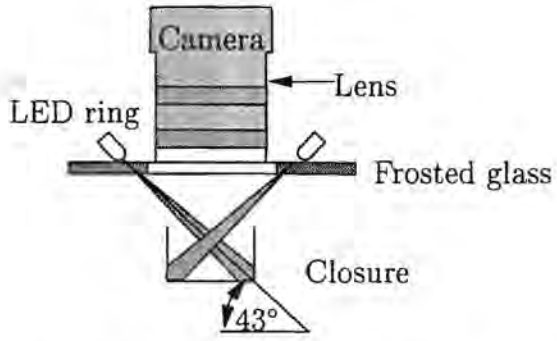



Figure 5.3: The layout of the LED lighting system.

that area is good. As the light is focused, it is not possible for light to shine directly on the lens. Despite the disadvantages, the fact that the LED lighting system is more robust and is compact makes this system suitable for use in the factory environment.

## 5.3 The camera and camera positioning

The camera is the device which converts the optical image into a signal. This is done using a CCD camera which provides a video signal as its output. The rate at which closures pass the camera is an important consideration in the choice of a camera. In order to see into the closure and to see the complete liner boundary, the closure must be directly below the camera when the image is taken. The normal frame rate of a video camera is 25 frames per second. The closure rate is 20 per second. These rates are not the same and the closure rate cannot be adjusted to be synchronous to the frame rate. As a result it is not possible to use a normal synchronous video camera as the image of the closure cannot be taken at a specified time.

An asynchronous video camera, one which can be triggered using an external signal, is needed. A Pulnix $TM620$ camera was chosen. The data sheet for the camera is included in Appendix C. The shutter speed can be adjusted from $\frac{1}{50}s$ to $\frac{1}{10\ 000}s$. This is suitable in terms of the shutter speed chosen in Section 3.6. The camera has good low light sensitivity, 0.5 $lux$, this enables the high shutter speed to be used.

### 5.3.1 Camera temperature

In order to keep the camera below its maximum operating temperature while operating with the incandescent lighting system, forced cooling was needed. This was done using compressed air as this was readily available on site.

With the LED lighting system, the ambient temperature is low enough that forced cooling was not necessary. This reduced the amount of dirt brought into the optical system. This is important as it reduces the maintenance required.

## 5.4 Host computer

The host computer is the central part of the image processing system. Its functions are discussed in detail in Chapter 7. The computer is a 486 $DX2$ running at 66 $MHz$. This supplies power to the DSP and the frame grabber and the ISA bus is used for communication between the DSP and the main processor. The I/O card described in Section 5.7 supplies the I/O requirement for the computer from sources other than the DSP.

## 5.5 DSP processor board

The DSP processor board has a Motorola DSP 56001 processor running at 33 $MHz$. There are two buffers for image storage. This allows for one image to be processed while the second image is digitised and stored in the other buffer ready to be processed. This method is necessary due to the rate at which the closures pass through the system. A video signal from the camera takes 50 $ms$ to be digitised. This is approximately the same time it takes for one closure to pass through the system, thus it is necessary for one frame to be processed while the next frame is digitised.

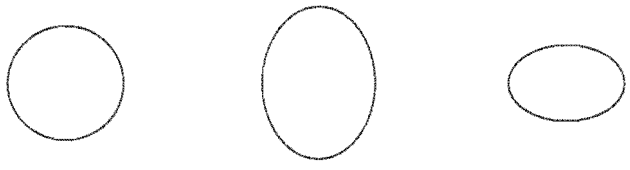
## 5.6 Frame grabber

The function of the frame grabber is to digitise the analogue video signal from the camera and store it in the appropriate frame buffer. The camera trigger signal is also received by the frame grabber to indicate when to start digitising the signal.

The video signal from the camera is an interlaced image. Each field is stored in alternate lines in the frame buffer. Due to the movement of the closure, only one field from each frame can be used for processing. This is due to the movement of the closure in the image between fields, resulting in the fields being offset by a few pixels in each frame.

The video signal is a PAL signal. The image has a 4/3 aspect ratio. The digital image digitised by the frame grabber is 512 rows $\times$ 512 columns. If a circle is grabbed it will appear elliptical with the semi-major axis in a vertical direction. Due to the movement of an object

in the image only a single field is used, thus the image used is 256 rows × 512 columns. Thus the circle now appears as an ellipse with the semi major axis in the horizontal direction. This means that the resolution of the image is not equal in the horizontal and vertical directions. Figure 5.4 shows the three stages described.



(a) Original shape of object. (b) Object shape in frame. (c) Object shape in a single field.

Figure 5.4: Showing an original object with its distortion due to the aspect ration of the pixels. The final image shows the expected shape of a bottle closure in the processed image.



## 5.7 Input/Output card

The input output card(I/O card) is the interface between the host computer and external components of the system other than the frame grabber and the DSP processor card. Signals which need to be received by the host computer are

**Encoder signal** This is an interrupt signal consisting of pulses which are synchronous to the rotation of the lining machine. This is used for tracking the position of the closures through the system, Section 5.7.1

**Detector signal** This indicates the presence of a closure in the system, Section 5.7.2.

**Temperature** This was used with the incandescent lighting system to signal if the camera temperature was too high, Section 5.7.3.

**Light integrity check** An alarm condition was generated if one of the incandescent lights failed. This signal became redundant with the LED lighting system, Section 5.7.4.

Two output signals are supplied

**Eject signal** This signal is used to eject defective closures from the system, Section 5.7.5.

**Camera trigger** This signal is the required signal for the camera to take an image.

A schematic of the I/O card is shown in Appendix A.

### 5.7.1 Rotary encoder

The rotary encoder is used to supply a ticker interrupt which is synchronous to the rotation of the lining machine. This interrupt is used to provide the timing of the system so that the progress of each closure through the system can be tracked. A description of how this is used is given in Chapter 7.

### 5.7.2 Closure detector

The closure detector is used to indicate the presence of a closure passing through the system. An Omicron E3S-X3CE4 fibre optic sensor is used. There are two fibres, the first transmits light and the second receives reflected light. If a closure is in front of the detector the signal changes state until the closure has passed.

### 5.7.3 Temperature sensor

The temperature sensor is used to monitor the camera temperature. A simple system has been used to set a signal high if the camera temperature increases above a preset limit. If this limit is reached, an alarm is raised and an operator can check the camera to prevent damage due to overheating.

The temperature is sensed using a single chip window comparator chip $TMP01$ on which the temperature is set to a user defined value. Four resistors are used to set the minimum and maximum temperature. In this system only the maximum temperature setting is needed. The temperature sensor is mounted directly onto the camera.

### 5.7.4 Light integrity check

The correct operation of the lights is fundamental to the operation of the system. Due to reliability problems with the incandescent lighting system it was necessary to monitor the bulbs. If one of the bulbs was not working, the results of the processing were meaningless. The current supplied to the globes was monitored using a sense resistor and a comparator. The four 50 $W$ bulbs drew 16 $A$, the voltage drop across a .33$\Omega$ resistor was monitored using

a comparator. If the voltage across the resistor dropped by one volt the comparator changed state and the signal monitored by the I/O card changes and the host computer raised an alarm condition.

The monitoring of the lighting became redundant when the LED lighting system was implemented as the LEDs have a much longer life.

### 5.7.5 Ejection system

A jet of compressed air is used to eject defective closures. This is done using a Festo $MFH-3-M5$ solenoid valve which is controlled by the host computer.

The switching of the solenoid valve injects noise back onto the power supply. This noise interfered with the other signals. The noise was suppressed to a tolerable level using a zener diode as a simple snubber.

The hardware provides a platform on which the system can be developed. Chapter 7 explains how the different hardware components and the software are combined to form the overall inspection system. Before the overall system can be discussed, the image processing algorithms which are used to detect defects need to be presented. This is done in the following chapter.

# Chapter 6

# Image Processing Algorithms



## 6.1 Introduction

This chapter covers the various image processing algorithms that were developed to determine if a closure had a defect on the liner or not. Sections 6.2 and 6.3 discuss aspects relating to the efficiency and stability of the algorithms. These are factors which have to be considered when developing an automatic system which has to run in real-time.

Section 6.4 discusses methods of designing and using a global threshold to segment the liner from the image. Section 6.5 presents ways of estimating the position of the liner boundary based on known information and making small searches in the image. This can be used to restrict the amount of processing of the image.

The next part of the chapter gives three different methods of finding the liner boundary. The first uses a boundary tracking algorithm which tracks along the strong edge of the liner boundary, Section 6.6. The second uses a method of relaxing a known elliptical approximation of the boundary onto the actual boundary, Section 6.7. The third method used a threshold to extract the boundary from the image and then to use a binary boundary tracker to extract the boundary, Section 6.8.

Section 6.9 describes the algorithms used to find the defects from the boundary information. Flashing is detected by examining the symmetry of the boundary. Non-fills are detected by examining the pixel intensities along the boundary.

The rest of the chapter presents methods of finding non-fills using grey-scale morphology. This includes using a top-hat transform, Section 6.13 , and morphological gradients, Section 6.14.

## 6.2 Processing efficiency of algorithms

The processing time available per closure is limited to about 50 $ms$ by the rate at which the closures are produced. Thus the design of any algorithm must take this into consideration.

There are three primary factors to be considered when designing an algorithm to run in a limited time frame. These are

- The amount of data to be processed. In this system it relates to the number of pixels in an image that need to be processed. Each pixel access takes time to complete. Repeated accesses of the same pixels can be thought of as increasing the amount of data.
- The operations applied to the data. More complex operations, such as a division, will take longer to complete than simple operations, such as addition.
- The number of times an operation is applied to the data. A simple operation repeated many times takes longer than a single more complex operation.

The strategy taken was to keep the processing efficiency needed in mind when designing the various algorithms. However the first criteria is to design an algorithm which will work. The second step is to determine if it can be implemented on a specific hardware base and actually be implemented in the time available. It is important not to give up on an idea because it may not be able to be implemented fast enough. On the other hand processes which are obviously too computationally intensive should be avoided. If every idea is abandoned due to it being too slow to implement then one may end up with no solution. If a solution can be found but is too slow to implement, work can be done on choosing new hardware or methods which will enable the solution to be implemented.



## 6.3 Stability of algorithms

This section examines various criteria which need to be considered to ensure that the system remains stable while running the algorithm. This involves ensuring that there is no possible way for the algorithm to get into an infinite loop or to follow a path that is obviously too long. This may involve checking for special cases. Prior information about the object is a very efficient method of monitoring the state of the algorithm and deciding if it is behaving in a normal way.

The algorithms were designed using limited test data which did not include all the possible images. The algorithm may get lost while processing some images. The algorithm needs to

be able to cope with all these situations. This must be kept in mind in all further algorithm design.

## 6.4 Thresholding

This section discusses the method of using a single global threshold to segment the image to separate the liner and the closure. This provides the first stage for two algorithms which examine the shape of the boundary.

The liner and the closure are at different intensities in the image, see Figure 2.1. The closure appears as a bright area and the liner dark grey. Thus it is possible to use the differences in pixel intensities to label the liner and closure separately. In order to be able to accurately and reliably label a pixel as part of the liner, as opposed to being part of the closure, there needs to be a large variation between the grey-scale value of the liner and the grey-scale value of the closure. The smaller the difference in the grey-scale values the greater the chance of incorrectly labelling a point on the liner as part of the closure and vice versa.

A single global threshold may be used if the lighting is uniform and there is not an intensity gradient across the image. It is possible to design thresholding algorithms which take gradients across the image into consideration. This is done by using information from the image to decide on a local threshold. These methods are not suitable in this application due to the time constraint. Thus it is not possible to use a method which uses a lot of image information to determine a threshold.

A cross section through a closure is shown in Figure 6.1. It is clear that the characteristics necessary to use a global threshold are present. Firstly there is no gradient in the image of the liner. Secondly there is a significant difference in image intensity between the liner and the closure enabling a single threshold to be chosen. It is possible for a feature to be present in an image which may prevent the correct labelling of pixel as part of the liner. One of these is a bright spot near the edge of the liner which causes the liner to have a bright appearance. These shiny spots are caused by moisture from the compressed air which drives the tool which stamps the liner into the closure. The shiny spots may be labelled as part of the closure. An example of a shiny spot can be seen in Figure 6.1.

If a correct threshold can be chosen, a global threshold can be used to segment the image. This cannot be done with total reliability due to effects of shiny spots near the edge of the liner.

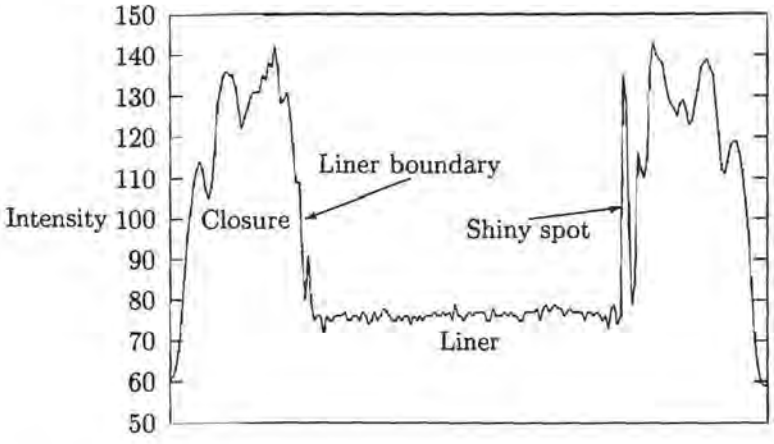



Figure 6.1: A cross section through the centre of a closure. This shows the typical grey-scale values of the liner and closure. The effect of a shiny spot on the liner can be clearly seen.

### 6.4.1 Choosing the threshold level

A correct threshold level needs to be chosen to separate the liner and the closure. This cannot be a fixed level for all images due to a variation in liner and closure intensity between images. The best method of determining the correct threshold level is to examine the entire image. The task of setting up a histogram is time consuming. As a result, a threshold level needs to be chosen based on a sample of a few pixels. This is based on the assumption that the sample pixel values give a good indication of the area they are representing.

The closure is positioned in the centre of the image, thus the approximate liner position is known. The liner intensity is not sampled near the centre of the liner as the lighting is more likely to have a bright spot or other artifact from the lighting here. The sample of liner intensity is taken from a region approximately half-way between the edge of the liner and the centre of the liner.

The number of pixels taken in the sample is a compromise between getting a good estimation of the liner intensity and speed. It would be best to take an average of every pixel in the liner. This is a problem as the position of the liner is not yet known. Based on the assumption that the liner intensity is the same, only a few pixels need to be sampled.

It was found that the average of a sample using an $8 \times 8$ block of pixels gave a good estimation of the liner intensity. Table 6.1 shows that using the average of 64 pixels gives a reasonable estimate of the average liner intensity. This table shows that unless the sample is affected by noise or shiny spot, a small sample will give approximately the same average as using an

average of all liner pixels. A group of 64 pixels is chosen as a smaller block of pixels is more easily affected by noise. Dividing by 16 can be implemented quickly by shifting the number right by 5 positions. Making the algorithm for determining the average intensity more robust involves using the history. If the threshold differs too far from the average threshold of the preceding 10 closures, the average is used. This is likely to be close to the correct threshold. This method works well if the liner intensity is smooth across the entire liner.

| Size of region | Average intensity |
|---|---|
| Whole liner | 76.8 |
| $64 \times 64$ | $77.1 \pm 1$ |
| $32 \times 32$ | $75.9 \pm 3$ |
| $16 \times 16$ | $75.8 \pm 4$ |
| $8 \times 8$ | $77.5 \pm 7$ |
| $4 \times 4$ | $80.2 \pm 11$ |
| $2 \times 2$ | $81.1 \pm 13$ |
| $1 \times 1$ | $80.3 \pm 15$ |

Table 6.1: A table showing the different values of average intensity obtained from sampling different numbers of pixels.

The liner may be shiny due to either the lining material being used or moisture on the lining material. This produces reflections which cause the liner intensity to appear uneven. This is particularly prevalent towards the centre of the liner. To avoid this effect, the liner intensity is not sampled near the centre of the liner. To sample closer to the edge of the liner involves knowing exactly where the edge is. If this first step is incorrect it follows that the threshold value determined will be incorrect. Three factors are used to determine where the edge of the liner is; (a) the history of previous edge positions (these are known as the boundary has been detected); (b) the limits of the liner edge determined during the calibration stage and (c) the edge strength and the expected liner pixel intensity values.

Once the intensity of the liner has been determined, the threshold can be set. The threshold is set at a percentage above the liner grey-scale intensity. If the image is thresholded at this level the liner is separated from the closure. The next stage is to locate the liner boundary which is done using two methods. The first is to use the known information about the shape of the liner as an initial estimate and to modify this based on the actual shape of the liner, Section 6.7. The second method is by using a binary boundary tracker, Section 6.8.

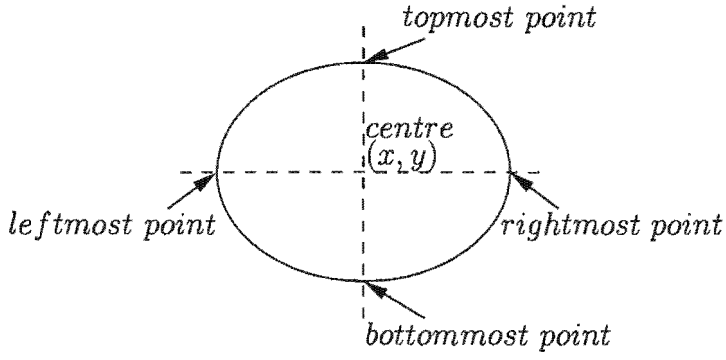



Figure 6.2: Finding the centre and the major and minor axes of the ellipse.

## 6.5 Estimating the position of the boundary

This section examines methods of using prior information about the shape and size of the liner to detect the position of the liner boundary.

The liner boundary being searched for has a known shape. In the case of a good closure the liner will be circular. A defective closure liner will generally be approximately circular with small deviations from this normal shape where the defect is.

The boundary of the liner is circular. This is distorted to an elliptical shape due to the aspect ratio, Figure 5.4. The distortion of the circle is only in the vertical and horizontal directions and thus the major and minor axes of the ellipse will always be parallel to the horizontal and vertical axes. With these limits on the rotation of the ellipse, four points are needed to define the ellipse.

In order to be able to define the ellipse it is easiest if the four extremal points are know. As the position of the liner in the image is approximately known it is easy to search for the four extremal points. From the extremal points, shown in Figure 6.2, it is simple to determine the centre point of the ellipse and the semi-major and semi-minor axes of the ellipse.

$$centre\ x = \frac{rightmost\ point + leftmost\ point}{2} \tag{6.1}$$

$$centre\ y = \frac{bottommost\ point + topmost\ point}{2} \tag{6.2}$$

$$major\ axis = \frac{rightmost\ point - leftmost\ point}{2} \tag{6.3}$$

$$minor\ axis = \frac{bottommost\ point - topmost\ point}{2} \tag{6.4}$$

The resulting ellipse is then determined using the algorithm in [9, page 88]. This is a fast algorithm for drawing an ellipse in a discrete image. If the four extreme points where correctly identified then an estimation of the ellipse to within 2 pixels can be found. Care must be taken when finding the midpoint of two integer numbers or the answers may be truncated instead of rounded. This is done correctly by adding 1 to the number before the division by two.

## 6.6 Boundary tracking

This boundary tracking algorithm was based on a heuristic graph searching algorithm used by Moon [17] to track the boundary of an apple. The graph searching algorithm was simplified to reduce computational complexity. The simplification was done by using prior information about the shape of the liner boundary.

The liner boundary is distinguishable by a sudden change in the grey-scale from the white of the closure to the grey of the liner. This rapid change in grey-scale is a strong edge in the image. This boundary tracking algorithm follows the strongest edge around the liner.

The start of the track is found by determining the strongest edge in the region where the edge of the liner boundary is expected. The end condition is satisfied when the next point is in the same row as the starting point, this is not necessarily the same column. The boundary is divided up into eight sections. Each section is related to the direction of the liner boundary in that section. The present direction specifies the expected direction of the next boundary point. The set of successor boundary points for each direction are shown in Figure 6.3. The transitions between directions in the track is governed by the slope of the previous section of the boundary. $dx$ and $dy$ are the change in the horizontal and vertical coordinates over the previous $n = 5$ steps. The transitions between the boundary sections occurs when the rule in Table 6.2 is satisfied.

| Transition | Rule | Transition | Rule |
|---|---|---|---|
| $0 \rightarrow 1$ | $dx > 1$ | $4 \rightarrow 5$ | $-dx > 1$ |
| $1 \rightarrow 2$ | $dx > -5dy$ | $5 \rightarrow 6$ | $-dx > 5dy$ |
| $2 \rightarrow 3$ | $5dy > dx$ | $6 \rightarrow 7$ | $-5dy > -dx$ |
| $3 \rightarrow 4$ | $dx < 1$ | $7 \rightarrow 0$ | $-dx > 1$ |

Table 6.2: Transition rules between directions for the boundary tracker based on edge strength.

Edge detection is done using a simple directional edge detectors as the direction of the edge is

known. Edge detectors are perpendicular to the current direction. The edge strength at each next pixel position is calculated as the sum of the three pixels outside the boundary minus the sum of the three pixels inside the boundary.

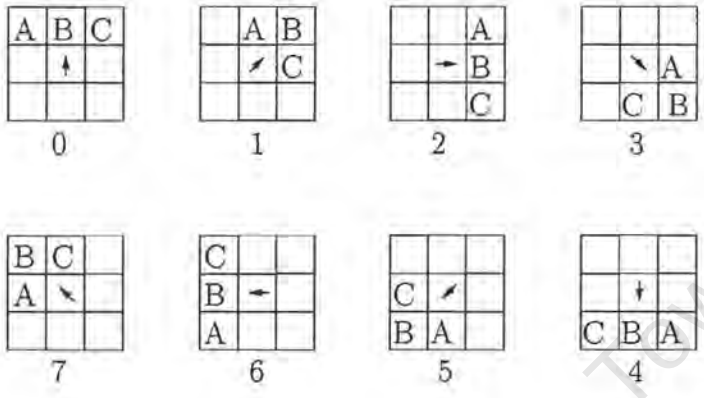



Figure 6.3: The expected next boundary point for the eight possible directions.

Starting at the determined starting point, the algorithm has the following steps

1. Calculate the edge strength at each of the three possible next boundary points.
2. The next point is assigned to the position with the highest edge strength.
3. If the new point satisfies the end condition, exit; otherwise continue.
4. Check transition rules and change state if necessary.
5. Go to 1.

### 6.6.1 Discussion of the edge strength boundary tracker

This method worked well yielding good tracks, particularly of the glass closures. This is because the liner boundary of the glass closures is very sharp and has a good edge strength, Section 2.3. This algorithm was implemented on the DSP with the classification algorithms in Section 6.9. The execution time of the algorithm on the DSP was approximately 48 *ms*. The result of a typical track is shown in Figure 8.1 with the resulting values for the symmetry measure and non-fill measure. Section 8.4 gives the inspection statistics from the production line.

In the case of the pet closures, this method of tracking the liner did not work. The track on the pet closures was often incorrect due to the liner boundary being indistinct and the colour

of the liner being lighter. The lighter colour meant that the edge strength was weak. As the track was poor, a valid result could not be obtained for these closures.

Other factors which made this algorithm unsuitable are that the edge detectors are differential operators which are very sensitive to noise in the image.

The results achieved for the inspection using this boundary tracker on the glass closures were well below the required specification. It was not possible to use it to inspect pet closures. It was decided that a different method was needed to detect the liner boundary. Two more methods are given in Sections 6.7 and 6.8.

## 6.7 Curve relaxation algorithm

Methods of finding a boundary in an image using prior knowledge of the shape are, boundary tracking, Section 6.6, region growing and line detection algorithms. Gritton and Parrish [10] suggest finding the boundary points of a cell by using an ellipse as an initial approximation and relaxing this onto the actual boundary points. This involves an initial approximation based on prior information followed by using image information to improve the approximation.

Methods of refining an estimated boundary onto the actual boundary are computationally intensive because the same task is repeated a number of times on similar data. The closer the initial approximation is to the actual object the more efficient the processes is.

The initial approximation can be found by various methods and Section 6.5 provides a reasonable approximation of the actual curve. The approach of the algorithm from this point involves relaxing this boundary onto the actual boundary in the image. During this process the boundary must remain closed, that is, the moving of the boundary points onto the real boundary must not form gaps in the boundary. To do this a set of rules needs to be imposed on the movement of any boundary point for any pass.

- A point must be moved onto a better point if one exists. A better point is one with a higher edge strength where the image intensity is close to the average liner intensity.
- The boundary point must remain eight connected to its adjacent boundary points. If this is not satisfied, the boundary point must not move.

This operation is repeated until there is no further movement of a boundary point or a maximum number of passes are made. The resulting boundary is a better approximation of the actual boundary than the original estimate. This method involves testing the edge

strength and image intensity around the boundary a number of times. This proves to be computationally expensive due to the repeated access of pixels.

### 6.7.1 Discussion of the curve relaxation method

The advantage of this method is that unlike a boundary tracking method the boundary can be predefined using prior knowledge of the expected shape. This ensures that the final boundary is a good approximation of the actual liner boundary. A boundary tracker may give a boundary, that is, very different to the liner boundary as only local pixel information is used, and no prior information about the overall shape.

This algorithm gives a very good result but is computationally inefficient because of the repeated access of pixels. As a result it was not possible to implement it on the hardware as it will not run fast enough.

The ability to specify the boundary shape before the algorithm is implemented reduces sensitivity to noise. This is done by maintaining a connected boundary through successive iterations of the algorithm. This method was not implemented on the DSP as, from timing the prototype stages it would not be possible to implement it fast enough.

## 6.8 Finding the edge of the liner in a binary image

This method finds the boundary of the liner from a binary image. The binary image is obtained using the threshold method in Section 6.4. The liner is easily extracted using the threshold method. Once the liner has been extracted, the edge of the liner needs to be located. Two ways to do this are described below, the most computationally efficient method was found to be a binary boundary tracker.

### 6.8.1 Using binary morphology

Binary morphology can be used to find the boundary, the method is described in Section 4.5.1. The external boundary is being searched for, thus the image is subtracted from the dilated edge, this yields a binary image of the boundary. This method is only guaranteed to yield a single connected boundary if the original binary image is a single area with no holes. This method is not an efficient method computationally as it requires multiple accesses of all of the image pixels.

### 6.8.2 Using a binary boundary tracker

The previous methods require all the pixels in the image to be repeatedly accessed. A more efficient method of locating the boundary needs to be located while accessing a minimum number of pixels. An efficient method of finding the edge of the binary object is to track around the outside of the object. This method is guaranteed to find a single connected boundary around the outside of the liner in the image. A connected boundary is one which starts at a point, goes around the object and returns to the point where it started.

The first stage of the algorithm is to find the starting point. This point is also the end point. It is assumed that the boundary does not touch the edge of the image, thus no tests are made to take care of special conditions at the edge of the image. This assumption can be made based on known information about the problem. The tracker tracks around the boundary in a clockwise direction.

The boundary binary tracking algorithm works by examining pixels around the current pixel. This examination is done in a clockwise direction until the next boundary pixel is the same point as the starting pixel. The track is now complete.

There are special cases which need to be considered by the algorithm to ensure it is robust. If the threshold level is too high the binary image is such that the tracker has the potential to take a long route which would not yield a meaningful result as well as taking too long to complete. If the threshold is too low then the track will be too short if it is possible to find a start at all.

### 6.8.3 Finding the start of the binary track

An important part of the boundary tracker is finding the start position. The usual method of finding a start is to start on the closure area of the image and search inward towards the liner until the first pixel which is labelled as part of the liner is found.

This method generally yields the correct start. Cases where this is not so need to be addressed in order to make the algorithm robust. These are

- If a single pixel is found just outside of the liner the track will be one pixel long.
- If the threshold is too high the binary image will not represent the liner and the track will be too long.
- If the threshold is too low the track binary image will not represent the liner and the track will be too short.

- If the closure has moved slightly in the image and the starting point is inside the liner, that is it is not a boundary point, then the algorithm will not track.

If any of these problems are detected, action can be taken to rectify the problem. If the track is too short then a second attempt can be made at finding a better starting point. However if the track is too long there is not sufficient time to re-track the boundary. As the probability of the closure being good is very high, if no valid result can be returned, the closure is accepted.

The boundary that is found portrays different information about the closure. This can be used in different ways to determine these characteristics.

### 6.8.4 Discussion of the binary boundary tracker

This method worked well for tracking both glass and pet closures. The execution time of the algorithm on the DSP was approximately 49 *ms*. The results of the inspection of closures on the production line are shown in Section 8.5.

This algorithm is not affected by noise as much as the tracker based on edge strength. The two boundary trackers may appear to be very similar methods of determining the liner boundary, however they differ in a fundamental way. The method based on edge strength simultaneously attempts to extract the boundary and at the same time determine its position. The disadvantage 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 binary boundary method first extracts the liner using the threshold and then tracks the liner boundary. This method is affected less by noise as the segmentation is done separately, before the boundary position is determined. The threshold level is determined by averaging a number of pixels. The averaging operation is an integrating effect which reduces the effect of noise.

## 6.9 Checking for defects

This section examines the methods used to check for the presence of non-fills and flashing. The boundary has been found using the methods described in Sections 6.6, 6.7 and 6.8. This boundary portrays information about the shape and the exact position of the boundary. The shape information can be used to detect flashing and the image intensity at the boundary points can be used to detect non-fills.

### 6.9.1 Detecting flashing

Two stages are used to detect flashing. The first is to analyse the shape of the boundary and the second is to use a profile of the pixel intensities outside the boundary, as a check.

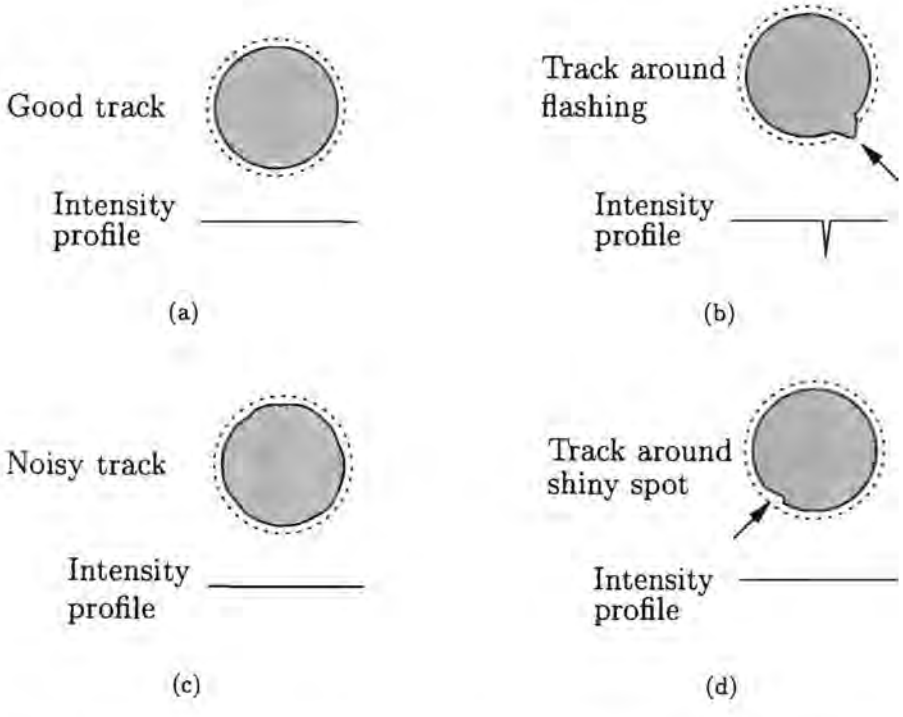



Figure 6.4: Four possible cases which need to be considered when classifying a closure for flashing. The closure liners are shown with an intensity profile, taken along the dotted line around the track, a few pixels from the liner boundary.

The first stage involves an analysis of the shape of the boundary. This is done as a flash will result in a distortion of the boundary shape. For an ideal boundary, the top and bottom halves of the boundary are mirror images. The same is true of the left and right halves. A symmetry measure is used to determine the degree that an actual boundary deviates from this.

This is done by determining the extent of the liner by finding the maximum and minimum row positions in each column. The axis of symmetry is from the maximum and minimum row of the boundary, $c_{max_i}$ and $c_{min_i}$. The symmetry measure in the vertical direction, $S_{vertical}$ is given by

$$S_{vertical} = \Sigma_i \left[ c_{max_i} + c_{min_i} - \overline{(c_{max_i} + c_{min_i})} \right]^2$$

A similar measure $S_{horizontal}$ can be found in the horizontal direction comparing the left and right halves. Due to the difference in resolution in the horizontal and vertical directions due to the aspect ratio distortion, a correction has to be made to ensure $S_{horizontal}$ does not dominate the symmetry measure. The total value for the symmetry measure is

$$S = \frac{4}{3}S_{vertical} + S_{horizontal}$$

The problem with this measure are: if the boundary track is noisy, as in Figure 6.4(c), or the tracker tracks around a shiny spot, as in Figure 6.4(d). In either of these cases the symmetry measure may be too large. As a result of this a second check is needed.

The second check involves using an intensity profile of a ring of pixels a few pixels away from the boundary. The circle on which these intensity profiles are taken are shown in Figure 6.4 as well as the profile. If an intensity profile crosses a flash this will cause a dip. This dip can be detected using a simple threshold. Thus if there is a high value for a symmetry measure but there is not a corresponding flash in the intensity profile then the closure is probably good. This is shown in Table 6.3. The method of using an intensity profile is not sensitive enough to use on its own.

| Image | Symmetry measure | Intensity profile | Classification |
|---|---|---|---|
| Figure 6.4(a) | Low | Smooth | No Flash |
| Figure 6.4(b) | High | Spiky | Flash |
| Figure 6.4(c) | High | Smooth | No Flash |
| Figure 6.4(d) | High | Spiky | No Flash |

Table 6.3: Showing the classification of closures for flashing based on the symmetry measure and the intensity profile.



To detect the presence of flashing in a closure, the appropriate threshold needs to be set for the symmetry measure which divides good closures from defective closures.

### 6.9.2 Detecting non-fills

The presence of a non-fill in an image is indicated by a slightly darker patch near the edge of the liner. Thus the algorithm to detect non-fills looks at the pixel intensities along the edge of the liner. The darkest pixel of the non-fill is often just inside the detected liner boundary. The intensity profile is determined by using the minimum pixel value of the boundary pixel and two pixels inside this pixel.

A non-fill appears as a sharp drop in the intensity profile. In order to detect this non-fill the

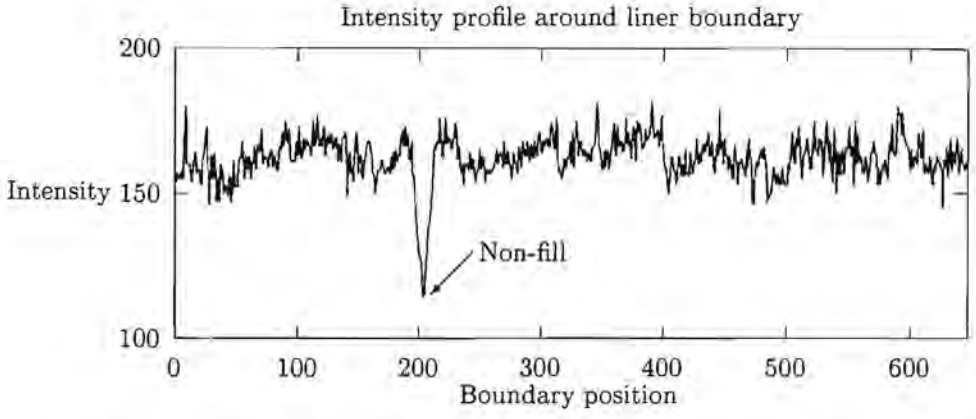



Figure 6.5: A typical intensity profile of a liner boundary showing a non-fill.

variance in a $2r_v + 1$ moving window is determined along the length of the signal and the maximum variance is stored. This maximum variance is used as a measure to test if there is a non-fill. If this maximum variance is above a set threshold then the closure is said to have a non-fill. If the maximum variance is below the threshold then the closure does not have a non-fill.

## 6.10 Using morphological filters to find non-fills

An image of a bottle closure has specific characteristics, see Sections 2.3 and 2.4. If the closure has a non-fill, this will appear as a change in the normal structure within the image. It is possible to use morphological filters to determine if a particular structure is present in an image. This information can be used to determine if the closure being examined has a non-fill.

## 6.11 Processing efficiency of morphological algorithms

The two basic operations of morphological image processing are erosion and dilation, Section 4.6. These involve comparing the pixels in an image to a structuring element. For each point in the structuring element, the test involves a subtraction and a comparison. This happens at every position in the image being examined. Because the region where the defects are is known, the computational time can be reduced by applying morphological filters in the region around the liner boundary only.

## Section 6.11: Processing efficiency of morphological algorithms

### 6.11.1 Reducing the search area

The only way to improve the efficiency of morphological processing algorithms is to reduce the number of pixels where the structuring element probes the image. This can be done in this case as the defects in the closure only occur on the liner boundary. The method described in Section 6.5 can be used to estimate the position of the liner boundary. If a band of pixels around this boundary is taken then any defect will be found in this band. Thus the morphological filters can be applied in this band. The image is $512 \times 256$ pixels. A band 16 pixels wide is chosen around the estimated boundary, the search area is reduced by about 93%.

There are important factors to note regarding this area of interest. Firstly it must be wide enough to ensure that the structuring element can be contained in the region. Secondly the band of pixels must be wide enough around the estimated boundary to ensure the actual boundary will be in the region of interest.



### 6.11.2 Selecting the size of the structuring element

There are three factors to consider when choosing the size of the structuring element to be used in this problem. Firstly, the size of the structuring element used to probe the image needs to be such, that it can detect the size of the defect being searched for. Secondly, the size of the structuring element is directly proportional to the processing time. Thirdly, as the search area is limited, the structuring element cannot be wider than the band of the region of interest, as image data outside this region of interest does not exist.

To ensure that a narrow region of interest can be used, there are certain cases where the repeated application of a smaller structuring element can yield the same effect as a single application of a larger structuring element. This includes the class of structuring elements which are flat, that is, all points in the structuring element are the same value. These structuring elements may be able to be decomposed if the original, larger structuring element, can be made up of the dilation of the two smaller structuring elements. Then using Equation 4.2, eroding the image by each of the structuring elements in turn will yield the same result as the erosion of the image by the original structuring element.

Thus, assume there is a structuring element $k$ which can be decomposed into two smaller structuring elements $g$ and $h$ such that

$$k = g \oplus h$$

then

$$f \ominus k = f \ominus (g \oplus h) = (f \ominus g) \ominus h$$

Thus eroding $f$ by $g$ and then by $h$ yields the same result as eroding $f$ by $k$.

A similar equation holds true for dilation, but has an extra degree of freedom due to the associative property of dilation. If

$$k = g \oplus h$$

then

$$f \oplus k = f \oplus (g \oplus h)$$

and then using the property that dilation is associative, Equation 4.3, it is seen that

$$f \oplus (g \oplus h) = (f \oplus g) \oplus h$$

The effect of splitting a structuring element into two smaller structuring elements has two advantages. The first is that fewer points are needed in the region of interest which can be narrower. A second is that it can lead to less processing. For example, if an erosion needs to be done with a $5 \times 5$ flat square structuring element with all its points zero. This would involve 25 comparisons and subtractions for each point in the region of interest. This same $5 \times 5$ structuring element can be made from the dilation of two $3 \times 3$ flat square structuring elements. Each of these erosions requires only 9 comparisons making a total of 18 comparisons for the overall operation. This is a saving of 7 or approximately 30% of the comparisons. This reduces the computation time accordingly. Often it is not possible to split a structuring element into different parts. However if simple structuring elements are chosen, this can lead to significant saving in computation time.



## 6.12 Morphological filters and the effects of nonuniform lighting

In Section 5.2 an image is described as being formed by the product of the illumination function and the reflective properties of the object. If the changes in the illumination function across the image are gradual then this variation will be removed by the morphological filters applied to the image. This is done without any prior knowledge of the illumination function.

If the illumination can be considered uniform in a local area then the undesirable effect of the non-uniform lighting is removed when the original image is subtracted from the processed image. This subtraction is used in both the top-hat transform, Section 4.6.3, and for

morphological gradients, Section 4.6.6.

## 6.13 Finding non-fills using the top-hat transform

In an image, a non-fill appears as a slightly darker dip in the image. If the image is inverted this will appear as a peak in the image. This peak can be detected using a top-hat transform operator, Section 4.6.3. To detect a non-fill involves using an appropriately sized structuring element.

The image is inverted to make the darker non-fill appear as a peak. The structuring element must be selected according to the size of the peak to be detected. The structuring element needs to be larger than the peak being search for. This is a problem in looking for non-fills as the size of the non-fill is not constant. The number of pixels of the non-fill is not known beforehand and can vary from very small, about $3 \times 3$ pixels, to very large, about $20 \times 20$ pixels.

### 6.13.1 Discussion of the top-hat transform

A compromise must be reached as to the maximum size of the non-fill that can be found and the computation time. As it is important to find all non-fills a large structuring element needs to be chosen. This needs to be of the order of 20 points. This means 400 comparisons for each position in the search which will make computation time excessive.

The top-hat transform did work to find non-fills. It was not successful in finding all the non-fills in the small test set. This means that it is not likely to achieve the success rate necessary for this problem. In general the results are very unsatisfactory. This means that this method cannot be used reliably to find all the non-fills.

The main reason why the top-hat transform is not able to detect non-fills reliably is that the non-fills lie next to a step edge in the image. The non-fills have a much smaller change in intensity than the change in intensity across the edge. This means that the non-fills are not always represented by a peak in the inverted image but only a variation in the slope, with a small local peak on the slope.

The second reason the top-hat transform fails is that the size of the structuring element does not match the size of the non-fill. As morphological processing uses specific shape structuring elements to probe the characteristics of the shape in the image. If the structuring element is not the correct shape then it cannot be used to accurately determine the image structure. This is a problem as the non-fills can take on any size and cannot be predetermined.

The third reason why the top-hat transform is not suitable is that the large structuring elements have a large computational overhead.

The results of the top-hat transform on the test images is given in Section 8.6.1.

## 6.14 Using a morphological gradient method to find non-fills

The non-fills occur at the edge of the liner. This is a region where there is a strong edge. A non-fill appears as a different rate of change in the boundary edge strength.

The output of a morphological gradient operator gives a grey-scale output where the largest grey-scale value is the steepest change in gradient. The result of this operation can be processed to determine if there is a non-fill along the edge of the liner or not.

A suitable structuring element needs to be chosen. This depends on a number of factors. The structuring element needs to be rotationally symmetrical as the boundary can have any orientation. The size of the structuring element affects the output gradient. If a small structuring element is chosen it has the effect of highlighting noise in the image. A large structuring element will cause a less distinct gradient to be found. Thus there has to be a compromise between the size of the defect that will be found and the noise rejection of the filter.



### 6.14.1 Finding a non-fill from the gradient information

Once the morphological gradient of the image has been taken, it needs to be checked for the presence of a non-fill. An example of this is shown in Figure 8.2. The non-fill appears as a dip in the intensity of the gradient information. Thus the non-fill can be detected using a threshold and tracking the result to check if it is a continuous boundary.

### 6.14.2 Discussion of the morphological gradient method

The result of using a $5 \times 5$ structuring element on a test set of 10 defective and 10 good closures is shown in Section 8.6.2. The success rate of this method is approximately 90%.

Although the results achieved using this method are reasonable, the algorithm cannot be implemented to run in 50 $ms$.

## 6.15 Discussion of morphological methods

The time available to implement any algorithm is limited to 50 *ms*. Although the morphological filters can be designed to find the defects in the image, it is not possible to implement them in the time available. At each point in the search space, a number of pixels in the structuring element have to be accessed. The result of this is that there are too many pixels in the image which need to be read from memory to be able to implement this algorithm in sufficient time.

The algorithms developed in this chapter enable defects in the images of the closures to be detected. The following chapter describes how these algorithms are combined with the hardware and the system software to produce the inspection system.

# Chapter 7

# Implementation

## 7.1 Introduction

This chapter covers how the hardware components in Chapter 5 and the different parts of the software work together to form the overall system. The linking of the hardware and the software is an integral part of the system. The implementation of the user interface and where the image processing algorithms fit into the system is discussed.

## 7.2 System overview

An overview of the complete system is given in this section. Figure 7.1 shows how the different hardware components of the system fit together. Arrows show the flow of information. A closure is said to be in the system when it is passing through the machine inspection system, this is from the time when a closure from the production line is first detected until the moment it passes the ejector.

The system is designed to operate in real-time, which means that a result for each closure must be produced before it leaves the system. If a result is only returned once the closure has passed the ejector then an action cannot be made based on the result. This means that there are 50 $ms$ available to process each closure.

The basic operation of the system on a single closure is covered in the following steps,

1. Detection of the closure as it enters the system.
2. Acquiring an image of the closure.

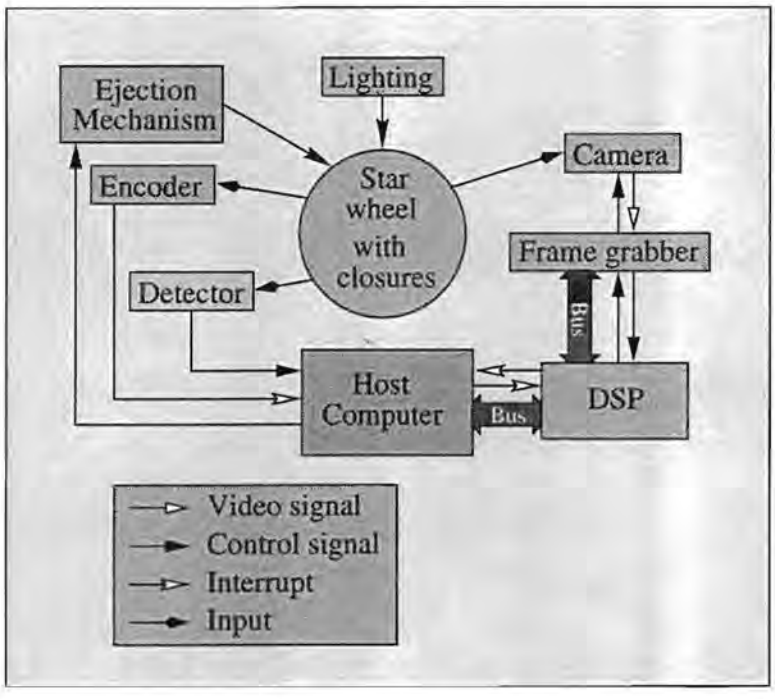



Figure 7.1: System block diagram.

3. Processing the image and returning the result to the host computer.

4. Ejecting the closure if the result indicates it is defective or allowing it through if it is not defective.

The above steps are complicated by a number of factors

- It takes approximately 50 $ms$ to digitise the image, this means that one image needs to be processed while the next image is digitised.

- There is more than one closure between the detector and ejector at the same time. This means the position of each closure in the system is important as the correct result must be associated with each closure to ensure that a correct classification is made.

- If a result is not returned for a closure or the result is returned too late then a closure cannot be classified.

### 7.2.1 System description

In this section the different components and their functions are shown in Figure 7.1 are described in more detail.

**Rotary encoder** The rotary encoder runs synchronously with the star-wheel and provides interrupts to the system. This is the primary method for timing in the system. On each interrupt the software checks to see if events have occurred . If an event has occurred they are placed in a queue by the interrupt routine and these events are processed between interrupts.

**Lighting** The details of the physical characteristics of the lighting are described in Section 5.2. The function of the lighting is simply to illuminate the closure to ensure that a good image is obtained.

**Detector** Detects the presence of a closure as it passes. The position of the closure is noted in terms of the current interrupt count. The position is tracked using the rotary encoder.

**Ejection mechanism** Defective closures are ejected from the system when they reach the ejector. A closure will reach the ejector a known number of interrupt pulses, from the rotary encoder, after it is detected.

**Camera** The camera provides a source for the images. A trigger needs to be sent to the camera when there is a closure beneath it to ensure good positioning of the closure in the image. A closure is beneath the camera a known number of interrupt pulses after it is detected. The trigger is supplied by the DSP.

**Frame grabber** The frame grabber digitises the image and places it into the memory of the DSP processor. When the DSP issues a trigger to the camera, the frame grabber receives the signal. When this signal is received the frame grabber knows to wait a predetermined time and then digitise the video signal. The frame grabber and DSP share their own bus thus communication is not via the ISA bus on the host computer.

**DSP card** The host computer hosts the DSP card and communication between the two processors is done using the ISA bus of the host computer. The digitised image from the frame grabber is stored in the DSP memory. The images are processed by the DSP and the results returned to the host computer.

## 7.3 Software

There are two separate programs which run independently on the host computer and DSP. The basic structure of the software on both processors is similar, this basic structure is shown in Section 7.3.2. The sections following this explain the software on the host computer and the DSP more thoroughly.

### 7.3.1 Event driven software

A simple procedural approach to the software was avoided to prevent the system from hanging if a procedure fails to complete. An event driven system was chosen as an alternative. When an interrupt occurs, the interrupt routine checks for events. Examples of events are a detected closure, a closure which has moved under the camera or a result ready to be logged. These events are placed in a queue waiting to be processed. The events in the queue are processed between interrupts. There are two advantages in using this system. Firstly if an event takes too long to process, it can be abandoned. In this way the whole system is not halted by a single incomplete process. The result for the abandoned process is lost. Secondly, less important events, such as logging results to a file, can be given a lower priority. This ensures that important events like processing an image or ejecting a closure, do occur. The lower priority events will only execute if there are no high priority events to process.

Figure 7.2 shows the implantation of the event driven software structure. The program is continually checking for events and processing them if they have occurred.



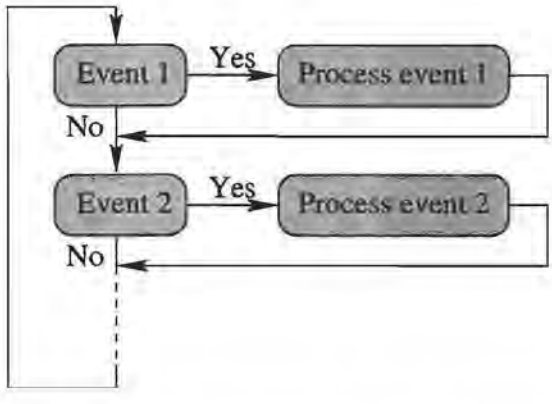



Figure 7.2: Implementation of event driven software.

### 7.3.2 Basic software structure

The software on both the host computer and the DSP have a similar basic structure which makes the maintenance easier. The two programs run independently but they need to communicate. Each program has a communication module which allows asynchronous communication between the two processors.

#### Host computer software

The host computer software modules and the flow of information between modules and the DSP software is shown in Figure 7.3. The modules are

**Main** When the program is started, this module runs the initialisation software on the host computer. During the program's operation, this module performs the main background processing loop. When the program is stopped this module terminates the process and exits.

**User interface** Provides an interface for the system operator and this displays the results of the processing. This module allows for the adjustment of system parameters such as the thresholds where classification decisions are made.

**Synchronisation** This module is responsible for the overall synchronisation and timing of the system. When a closure is detected, the interrupt count number is recorded. A known number of interrupt pulses later, the closure will be under the camera. At this time a trigger signal is sent to the camera to snap an image. This signal is simultaneously received by the DSP. Before the interrupt counter reaches the time when the closure is at the ejector, this module knows whether to eject the closure or not.

**Communication** This module is used to enable asynchronous communication between the host computer and the DSP. There are three types of communication. The first is the issue of an instruction from the host computer to the DSP. The second is the receiving of a result from the DSP, such as the feature values from the inspection. The third is the transfer of a stream of data for down-loading images from the frame grabber to disk. The communication is through the ISA bus of the host computer. Communication between the programs running on the two processors is achieved using a system of semaphores to synchronise the communication.

**Classification** Feature values of each inspected closure are returned from the DSP. The classification of the closure is done based on these features. If the closure is defective an instruction is sent to the synchronisation module to eject the closure when it passes

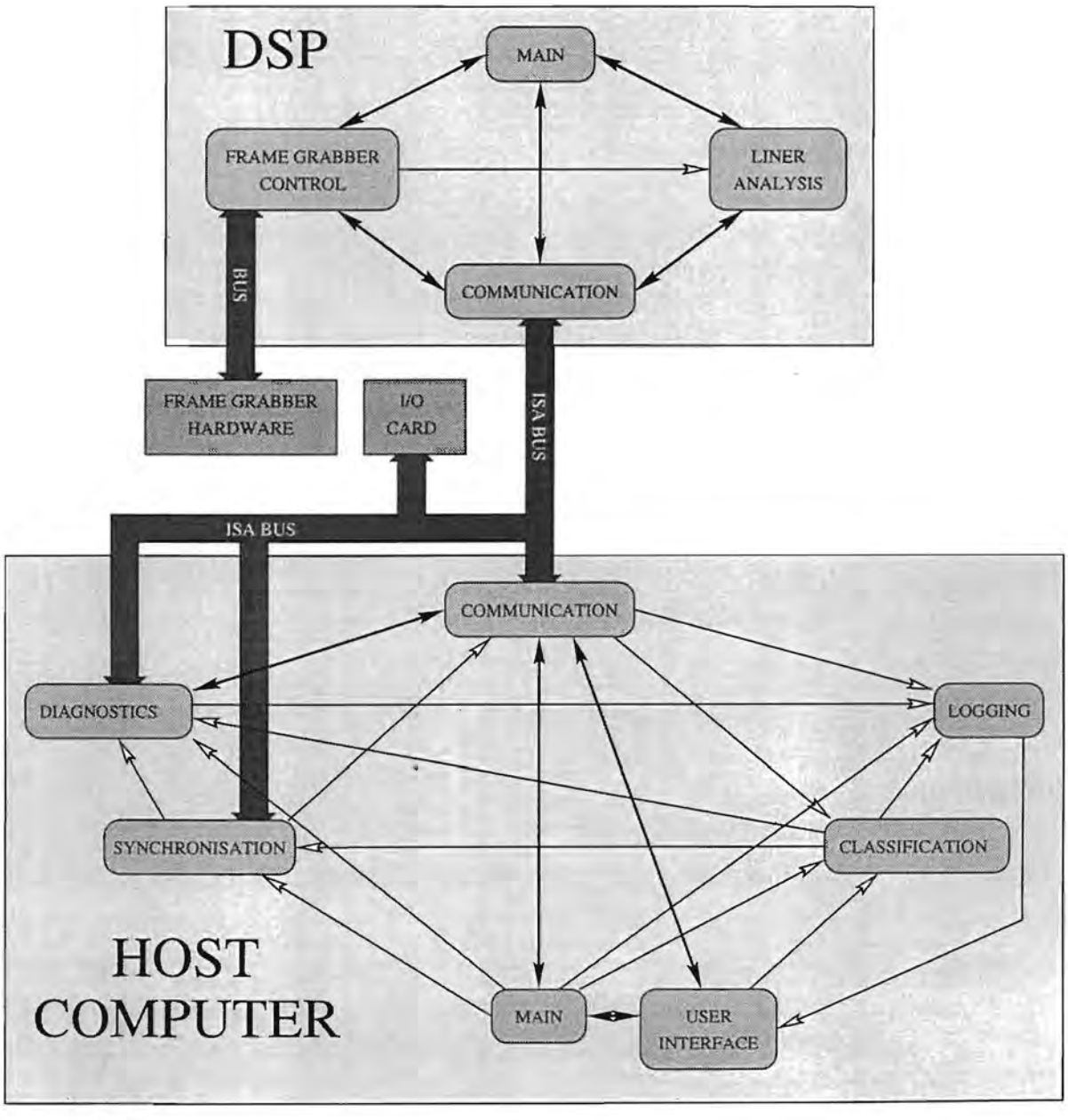



Figure 7.3: Structure and integration of software on the host computer and DSP.

the ejector. The results of the classification are sent to the logging module to be logged if required.

**Diagnostics** This module takes action if either a hardware or a software fault is detected by the system. The hardware signals from the I/O card are monitored. If the activity on these signals stops a fault is detected. A fault is also detected if the DSP stops responding to instructions from the host computer. If a fault is detected, the host computer first tries to rectify the fault by sending a reset signal to the appropriate hardware. If this fails to correct the problem, an alarm is raised to alert the operator.

**Logging** Information from the other software modules is available to be logged. This information is logged based on a user defined filter which decides what information is to be logged.

### DSP software

The DSP software has four modules. These are shown in Figure 7.3. The flow of information between the modules is indicated by the arrows. The modules are

**Main** On startup this module initialises the software modules on the DSP. Subsequently this module runs the main event processing loop. This main loop continually checks for a new image and issues instructions to the liner analysis module to process the image. This module controls the DSP during setup and calibration of the system.

**Frame grabber control** This module provides a high level interface to the frame grabber. This provides availability of instructions to snap an image or to down-load the image.

**Liner analysis** The image processing algorithms are implemented in this module. This includes the boundary tracking algorithm, Section 6.6 or 6.8 and the feature determination algorithms, Section 6.9. These results are returned to the host computer via the communication module.

**Communication** On start up this module establishes communication with the host computer. The instructions from the host computer are received by this module. The results of the processing are returned to the host computer.

## 7.4 Discussion of the system implementation

This system is to be implemented on a production line which works 24 hours a day. This means it must be reliable and not prone to stopping responding to signals or returning meaningless

results. As a result, many checks have been built into the system to take care of fault conditions. If a fault occurs which the host computer is not able to rectify, an alarm is raised to alert the operator.

The event driven approach to the software is a very effective method for this system. This allows an easy method to ignore low priority events if there is not enough time to execute them. The event driven software is effective when used in conjunction with the interrupt routines.

This chapter has given the implementation of the various components to provide a complete system. The next chapter presents the results of the inspection of the closures on the production line.

# Chapter 8

# Results



## 8.1 Introduction

This chapter presents the results of the work done. The first part presents the results of the boundary tracking methods of inspecting the liner boundary. These include the results of test runs on the system implemented on the lining machine. The latter part of the chapter presents the results using morphological filters to detect non-fills. These results are only for a set of test images as these algorithms were too slow the be implemented on the system.

## 8.2 An example of a detected liner boundary

If any of the methods of finding the liner boundary are successful then the results look very similar. The three possible methods are boundary tracking based on edge strength, Section 6.6, the curve relaxation method, Section 6.7 and the method of thresholding and tracking the resulting binary image, Section 6.8. The example track in Figure 8.1 shows the resulting liner boundary detected on a pet closure using the binary boundary tracking method. Using the algorithms in Section 6.9, the symmetry error is 1779 and the maximum variance of the intensity profile is 24. This symmetry error is typical of a good pet closure. A track on a good closure may give a symmetry error between 400 and 2500 depending on how noisy the track is. A closure with a flash may give any value for the symmetry error from 3000 for very small flashing to a maximum of 32 767, which is set by the algorithm, for very large flashing. The maximum variance is quite high for a good closure but is still in the range that is classified as good. The maximum variance measure may vary from about 8 to 25 for a good closure. For a defective closure this measure may be any value from 30 to 256. With

a typical thresholds on the symmetry error of 3000 and 30 respectively for the maximum variance, this closure will have been classified as good, which is a correct classification.

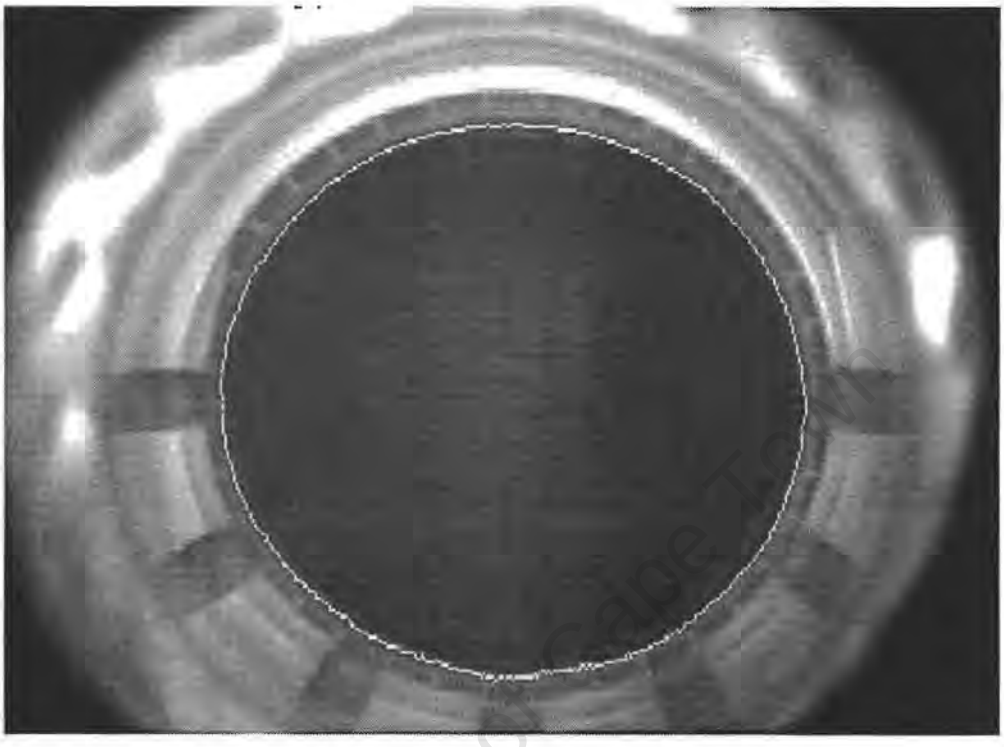


Figure 8.1: An example track on a pet closure.



## 8.3 Methods of generating results

The results were generated by using the system installed on the lining machine in the factory. To determine the statistics for a particular run, the ejected closures are collected while the system is running. The closures which are not ejected were examined for defects by two inspectors and the production supervisor. Any defects which were found in this stream of closures, which should have been only good closures, were removed and counted. Any other closures where assumed to be good. The ejected closures where manually inspected after the run, separating the good closures, the closures with flashing and non-fills.

These are represented in tables such as Table 8.1(a). On the left hand side of the table is the actual type of closure, good, flashing or non-fill. The top of the table lists the classification of each type of closure. If the closures where correctly classified, the diagonal positions in the table would all be 100%. Any percentages off the diagonal indicate the misclassification of closures. For example considering Table 8.1(a), 99.64% of good closures where classified

as good, 0.31% of good closures classified as flashing and 0.05% of good closures classified as non-fills.

## 8.4 Edge strength boundary tracker

The inspection system based on the edge strength boundary tracker, Section 6.6 worked on the glass closures only. Table 8.1(a) and (b) show the results for two runs. The third table, Table 8.1(c) shows the average result of the two runs. This shows the correct classification of good closures is 99%, flashing is 79% and non-fills is 88%.

| | Classification | | |
|---|---|---|---|
| | Good | Flashing | Non-fill |
| Good | 99.64 | 0.31 | 0.05 |
| Flashing | 20.33 | 79.67 | 0.00 |
| Non-fill | 22.22 | 0.00 | 77.78 |

(a) Run 1, 8144 closures

| | Classification | | |
|---|---|---|---|
| | Good | Flashing | Non-fill |
| Good | 98.87 | 0.52 | 0.61 |
| Flashing | 21.76 | 78.24 | 0.00 |
| Non-fill | 10.98 | 0.00 | 89.01 |

(b) Run 2, 20 620 closures

| | Classification | | |
|---|---|---|---|
| | Good | Flashing | Non-fill |
| Good | 99.08 | 0.46 | 0.46 |
| Flashing | 21.16 | 78.83 | 0.00 |
| Non-fill | 11.70 | 0.00 | 88.30 |

(c) Overall average result for both runs.

Table 8.1: Results of the inspection of bottle closures using the boundary tracking algorithm based on edge strength information.

The results of using the boundary tracking algorithm based on edge strength, for pet closures, are not shown. The algorithm typically rejected 50% of the good closures as flashing. If the decision level for the symmetry error was increased to prevent this high rate of rejection then the system failed to detect many of the closures with flashing.

The inspection system based on this algorithm achieved reasonable results for the inspection. The rejection of approximately 1% of good closures as defective may have been considered acceptable if the rejection of defective closures was better than approximately 80% which was achieved. The false rejection of good closures at the rate of 1% out of 2 million per day equates to the waste of 20 000 closures. Approximately 1.6% of the production in the above

test runs were defective, this equates to about 33 000 closures a day. With the rejection rate of defective closures at approximately 80%, 26 400 of these defective closures would be rejected while 6 600 would be accepted. Overall the result achieved is below the level required for quality control as specified in Section 2.5

## 8.5 Binary boundary tracker

The inspection using the binary boundary tracker to determine the position of the liner boundary, Section 6.8, was used successfully to inspect both glass and pet closures. The algorithm was only tested on the production line on pet closures. Test on glass closures were done on a set of 100 images taken previously on the production line. Tests could not be performed on glass closures on the production line as they were not being produced at the time of the tests.

The results for the glass closures are shown in Table 8.2. This shows the algorithm correctly identified all the closures. However there were no images with non-fills and the test sample is small compared to a run on the production line.

| | Classification | | |
|---|---|---|---|
| | Good | Flashing | Non-fill |
| Good | 70 | 0 | 0 |
| Flashing | 0 | 30 | 0 |
| Non-fill | 0 | 0 | 0 |

Table 8.2: Result of using the thresholding and using the binary boundary tracking algorithm to detect the boundary a set of 100 images of glass closures.

The results for the pet closures were generated from 5 runs, of between ten and twenty thousand closures, on the production line. The results of this inspection are presented in the tables in Table 8.3. The final table, Table 8.3(f), shows the average result. Of the good closures 99.89% were correctly identified. The waste generated by the false ejection of good closures is low, 0.1%, which equates to about 2000 closures out of a days production of 2 million. The rejection of defective closures is greater than 90%. Out of the test runs about 1.3% of closures where defective. This equates to approximately 26 000 defective closures per day and only 2 600 not being ejected.

Using this method the inspection results are good with low wastage of good closures and a reasonable rejection of defective closures. These results where not as good as the results specified in Section 2.5 but are much better than the results achieved using the boundary

| | Classification | | |
|---|---|---|---|
| | Good | Flashing | Non-fill |
| Good | 99.74 | 0.26 | 0.00 |
| Flashing | 8.57 | 91.43 | 0.00 |
| Non-fill | 14.29 | 0.00 | 85.71 |

(a) Run 1, 10 081

| | Classification | | |
|---|---|---|---|
| | Good | Flashing | Non-fill |
| Good | 99.80 | 0.20 | 0.00 |
| Flashing | 11.42 | 88.58 | 0.00 |
| Non-fill | 0.00 | 0.00 | 100.00 |

(b) Run 2, 10 077

| | Classification | | |
|---|---|---|---|
| | Good | Flashing | Non-fill |
| Good | 99.96 | 0.04 | 0.00 |
| Flashing | 12.46 | 87.53 | 0.00 |
| Non-fill | 0.00 | 0.00 | 0.00 |

(c) Run 3, 20 878

| | Classification | | |
|---|---|---|---|
| | Good | Flashing | Non-fill |
| Good | 99.85 | 0.14 | 0.00 |
| Flashing | 2.32 | 97.67 | 0.00 |
| Non-fill | 9.09 | 0.00 | 90.91 |

(d) Run 4, 12 393

| | Classification | | |
|---|---|---|---|
| | Good | Flashing | Non-fill |
| Good | 99.74 | 0.26 | 0.00 |
| Flashing | 7.73 | 92.27 | 0.00 |
| Non-fill | 0.00 | 0.00 | 0.00 |

(e) Run 5, 21 036

| | Classification | | |
|---|---|---|---|
| | Good | Flashing | Non-fill |
| Good | 99.89 | 0.11 | 0.00 |
| Flashing | 8.73 | 91.26 | 0.00 |
| Non-fill | 9.52 | 0.00 | 90.48 |

(f) Overall result for the 5 runs

Table 8.3: Results of the inspection of pet closures using the boundary tracking algorithm based on the threshold information.

tracker based on edge strength.

## 8.6 Grey-scale morphology

Two methods of detecting non-fills using grey-scale morphology, the first was using a top-hat transform, Section 6.13, and using morphological gradients, Section 6.14. These methods were not implemented on the DSP as they would not be able to run fast enough to be implemented on the production line. The results achieved by these methods were not very convincing and are presented in the following two sections.

### 8.6.1 Top-hat transform

The top-hat transform was applied to a set of 20 images of glass closures. The set consisted of 10 good closures and 10 closures with non-fills. The size of the structuring element affects the result. If the structuring element is too small then the output of the top-hat transform is noisy but the non-fill is highlighted. However the noise inhibits the ability to detect non-fills. If the structuring element is larger then the result is less noisy, however many non-fills are not detected.

The results were generated using three different size square structuring elements, $19 \times 19$, $11 \times 11$ and $5 \times 5$ pixels. The larger the structuring element the more the computation time. Table 8.4 shows the results for the 20 test images. With the $19 \times 19$ structuring element, only 4 out of 10 non-fills were detected with no good closures incorrectly classified. This structuring element is much larger than the smaller non-fills and is also wider than the liner boundary. The larger non-fill were correctly detected using this method. The very large non-fill, Non-fill 10, was not detected. Using a smaller $11 \times 11$ structuring element there was better detection of the non-fills with 7 out of 10 being correctly detected. This improvement came with a decrease in performance with 2 of the good closures being detected as having non-fills. Reducing the size of the structuring element further to $5 \times 5$ results in the output image being very noisy and non-fills are detected in almost all the images whether good or non-fill images.

The overall result of using the top-hat transform was not very effective. The size and shape of the structuring element has to be related to the size of the non-fill that is to be detected. The size of the non-fill is not known before the operation and there is no method of estimating its size. This results in an increased chance of a non-fill going undetected. If the structuring element is too small there are many false detections.

It is not possible to find a general structuring element that will detect all size and shape non-fills. The size and shape of non-fills are very variable thus the top-hat transform cannot be reliably used to detect these defects.

| Test Image | Structuring Element Size | | | Approximate size of non-fill |
|---|---|---|---|---|
| | 19 × 19 | 11 × 11 | 5 × 5 | |
| Good 1 | No detection | No detection | Non-fill detected | |
| Good 2 | No detection | Non-fill detected | Non-fill detected | |
| Good 3 | No detection | No detection | Non-fill detected | |
| Good 4 | No detection | No detection | Non-fill detected | |
| Good 5 | No detection | Non-fill detected | Non-fill detected | |
| Good 6 | No detection | No detection | No detection | |
| Good 7 | No detection | No detection | No detection | |
| Good 8 | No detection | No detection | Non-fill detected | |
| Good 9 | No detection | No detection | Non-fill detected | |
| Good 10 | No detection | No detection | Non-fill detected | |
| Non-fill 1 | No detection | Non-fill detected | Non-fill detected | 11 × 10 |
| Non-fill 2 | No detection | No-detection | No detection | 4 × 40 |
| Non-fill 3 | No detection | Non-fill detected | Non-fill detected | 9 × 12 |
| Non-fill 4 | No detection | No detection | Non-fill detected | 4 × 18 |
| Non-fill 5 | Non-fill detected | Non-fill detected | Non-fill detected | 12 × 10 |
| Non-fill 6 | Non-fill detected | Non-fill detected | Non-fill detected | 12 × 22 |
| Non-fill 7 | No detection | Non-fill detected | Non-fill detected | 8 × 30 |
| Non-fill 8 | Non-fill detected | Non-fill detection | Non-fill detected | 16 × 8 |
| Non-fill 9 | Non-fill detected | Non-fill detection | Non-fill detected | 12 × 22 |
| non-fill 10 | No detection | No detection | No detection | 28 × 40 |

Table 8.4: Results of top-hat transform using various size flat square structuring elements, followed by a threshold set at the maximum value in the image.

### 8.6.2 Morphological gradient results

The performance of this algorithm was tested on 20 test images of glass closures. The test set consisted of 10 images of good closures and 10 images of non-fills. The intensity of a typical gradient of a liner boundary is shown in Figure 8.2. The morphological gradient was only applied to the image in a narrow band around the expected liner boundary position. Table 8.5 shows the resulting classification of the 20 test closures. The correct classification of non-fills was 100% but 1 good closure was incorrectly classified.

The size of the structuring element does not affect the size of a non-fill that can be detected. This makes this method more successful than using the top-hat transform.

This method worked well at detecting non-fills. There was an incorrect classification of one of the good closures. The biggest problem with this method is that it does not give the size of the defect. This is a problem if the very small non-fills are to be accepted as good closures.

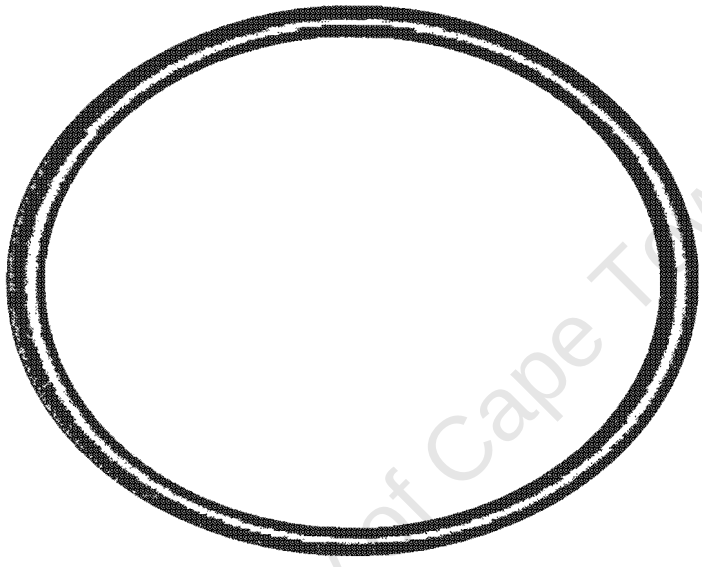


Figure 8.2: The typical output of a morphological gradient of a closure liner boundary using a $5 \times 5$ structuring element $g$ on an image $f$, $f - (f \ominus g)$.



| | Classification | |
|---|---|---|
| | Good | Non-fill |
| Good | 9 | 0 |
| Non-fill | 1 | 10 |

Table 8.5: The results of using the morphological gradient using a $5 \times 5$ square structuring element on a test set of 10 good closures and 10 closures with non-fills.

The morphological gradient method is fairly successful at detecting non-fills in closures. This method could not be implemented on the DSP to run within the available processing time. The structuring element needed is fairly small and thus is not excessively computationally intensive. The computation requirement was beyond the capabilities of the DSP used.

## 8.7 Discussion of results

Two basic ideas were pursued in finding defects in the closures. One was using grey-scale morphology. The second was using different methods to extract the liner boundary from the image.

Two morphology methods were tried, the top-hat transform, Section 6.13, and using morphological gradients, Section 6.14. The top-hat transform did not detect non-fill successfully, the size and shape of the structuring element needed to have a specific relationship to the size of the non-fill that was to be detected. Morphological gradients can be used successfully to detect non-fills. However this method was not implemented due to the restriction on the available time to process the image on the DSP.

Three algorithms were implemented to detect the liner boundary. The first based on tracking the boundary using edge strength information, Section 6.6, the second method relaxed a known ellipse onto the actual liner boundary, Section 6.7, and the final method segmented the liner from the image using a threshold and then tracking the resulting binary image to find the liner boundary.

The edge strength boundary tracker achieved reasonable results on the glass closures. This was unsuccessful on pet closures. The curve relaxation method worked well and always gave a good boundary but the implementation was too slow. The binary boundary tracking algorithm worked well on both pet and glass closures. The results using this method were about ten times better for the acceptance of good closures and twice as good for the rejection of defective closures than the boundary tracker based on edge strength.

The results achieved, using either of the two different boundary trackers implemented on the DSP, did not comply with the criteria specified in Section 2.5. The results using the binary boundary tracker were acceptable for use on the system. Further work must be done in order to improve the detection of the defective closures. Although the misclassification of the good closures is above the required level it is acceptable.

# Chapter 9

# Conclusions

## 9.1 Overview of work done

An automatic machine inspection system was developed to inspect plastic bottle closures. This system was capable of inspecting every closure in real-time at 20 per second. This means each closure is processed in 50 $ms$. The closures were inspected for defects which occur near the edge of the liner boundaries. These defects are non-fills and flashing. This inspection is to ensure that a high quality product is produced.

## 9.2 Discussion of results achieved

Two main types of algorithms were developed. One based on using grey-scale morphology to detect non-fills. The top-hat transform did not perform well. The size of the non-fill which was detected depended on the size of the structuring element. The method of using grey-scale morphology was successful. The size of the structuring element was not critical. The results based on a set of test data detected all the defects while only classifying 10% of the good closures as defects. The algorithms using morphology were not implemented on the DSP due to the processing required. As a result no results were generated on the lining machine.

The second type of algorithms used were those that detected the liner boundary. The symmetry of the liner boundary was used to detect the presence of flashing. The intensity profile of the liner boundary was used to detect non-fills. Three algorithms were developed to detect the liner boundary. The first was a boundary tracker which followed the strong edge around the liner. This performed reasonably well provided the edge was clear but the result could easily be disturbed by noise. The results for this were acceptable for glass closures but it did

not work on pet closures.

The second algorithm used an estimate of the expected boundary. This boundary was then relaxed onto the actual boundary based on edge strength and intensity. This worked well as it was insensitive to noise and used a more global scope of the image. The boundary detected was always a good approximation of the liner boundary in the image. This algorithm gave good results based on test images but was not implemented on the inspection system as it would take too long to process an image on the DSP.

The third algorithm, the binary boundary tracking algorithm described in Section 6.8, achieved the best results. The implementation of this algorithm to detect the liner boundaries and the algorithms to find the defects, described in Section 6.9 accepted 99.9% of the good closures and detected over 90% of the flashing and non-fills.

These results are not as good as those which were specified in Section 2.5. However they are considered acceptable to the manufacturer for the inspection of the bottle closures.

Further work needs to be done to reduce the number of defective closures that are accepted.

# Bibliography


[1] Herbert Boerner and Helmut Strecker. Automated x-ray inspection of aluminium castings. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-10:79–91, January 1988.

[2] John Canny. A computational approach to edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-8:679–698, November 1986.

[3] Josep R. Casas and Luis Torres. Coding of details in very low bit-rate systems. *IEEE Transactions on Circuits and Systems for Video Technology*, 4:317–327, June 1994.

[4] A.M. Darwish and A.K. Jain. A rule based approach for visual pattern inspection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-10:56–68, January 1988.

[5] Edward R. Dougherty. *An Introduction to Morphological Image Processing*, volume TT9 of *SPIE Tutorial Texts in Optical Engineering*. SPIE Optical Engineering Press, 1992.

[6] Edward R. Dougherty and Jaakko Astola. *An introduction to nonlinear image processing*, volume TT16 of *SPIE Tutorial Texts in Optical Engineering*. SPIE Optical Engineering Press, 1994.

[7] General Electric. Special purpose illumination supplies. Advertisement, March 1995.

[8] David S. Falk, B. Dieter, and D. Stork. *Seeing the Light Optics in nature, Photography, Color, Vision an Holography*. Harper and Row, 1986.

[9] J.D. Foley, A van Dam, S.K. Feiner, and J.F. Hughes. *Computer Graphics - Principles and Practice*. Addison-Wesley Publishing Company, 1990.

[10] C.W.K. Gritton and Edward A. Parrish. Boundary location from and initial plan: The bead chain algorithm. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-5:8–13, January 1983.

## BIBLIOGRAPHY


[11] Yasuhiko Hara, Hideaki Doi, Koichi Karasaki, and Tadashi Iida. A system for pcb automated inspection using fluorescent light. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-10:69–78, January 1988.

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

[13] Robert M. Haralick, Stanley R. Sternberg, and Xinhua Zhang. Image analysis using mathematical morphology. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-9:532–550, July 1987.

[14] B.K.P. Horn. *Robot Vision*. Mc Graw-Hill, 1986.

[15] Zahid Hussain. *Digital Image Processing: Practical Applications of Parallel Processing Techniques*. Ellis Horwood Series in Digital and Signal Processing. Ellis Horwood Limited, 1991.

[16] Petros Maragos and Ronald W. Schafer. Morphological systems for multidimensional signal processing. *Proceedings of the IEEE*, 78:690–710, April 1990.

[17] Peter Olaf Moon. Boundary tracking algorithm to delineate the boundaries of apple images. University of Cape Town, 1991.

[18] Peter Olaf Moon. An investigation into the design of a satellite based stereo imaging sensor and the use of automatic matching in the production of elevation models. Master's thesis, University of Cape Town, 1993.

[19] Williard D. Morgan and Henry M. Lester. *The New Leica Manual*. Morgan and Lester Publishers New York, 1953.

[20] Don E. Pearson and John A. Robinson. Visual communication at very low data rates. *Proceedings of the IEEE*, 73:795–812, April 1985.

[21] RCA Corporation. *RCA Electro-Optics Handbook*, second edition, 1974.

[22] Jean-François Rivest, Pierre Soille, and Serge Beucher. Morphological gradients. *Journal of Electronic Imaging*, 2(4):326–336, October 1993.

[23] Society of Manufacturing Engineers. *Inspection of Pharmaceutical Packaging with Linear-Array Video Sensors*, April 1989.

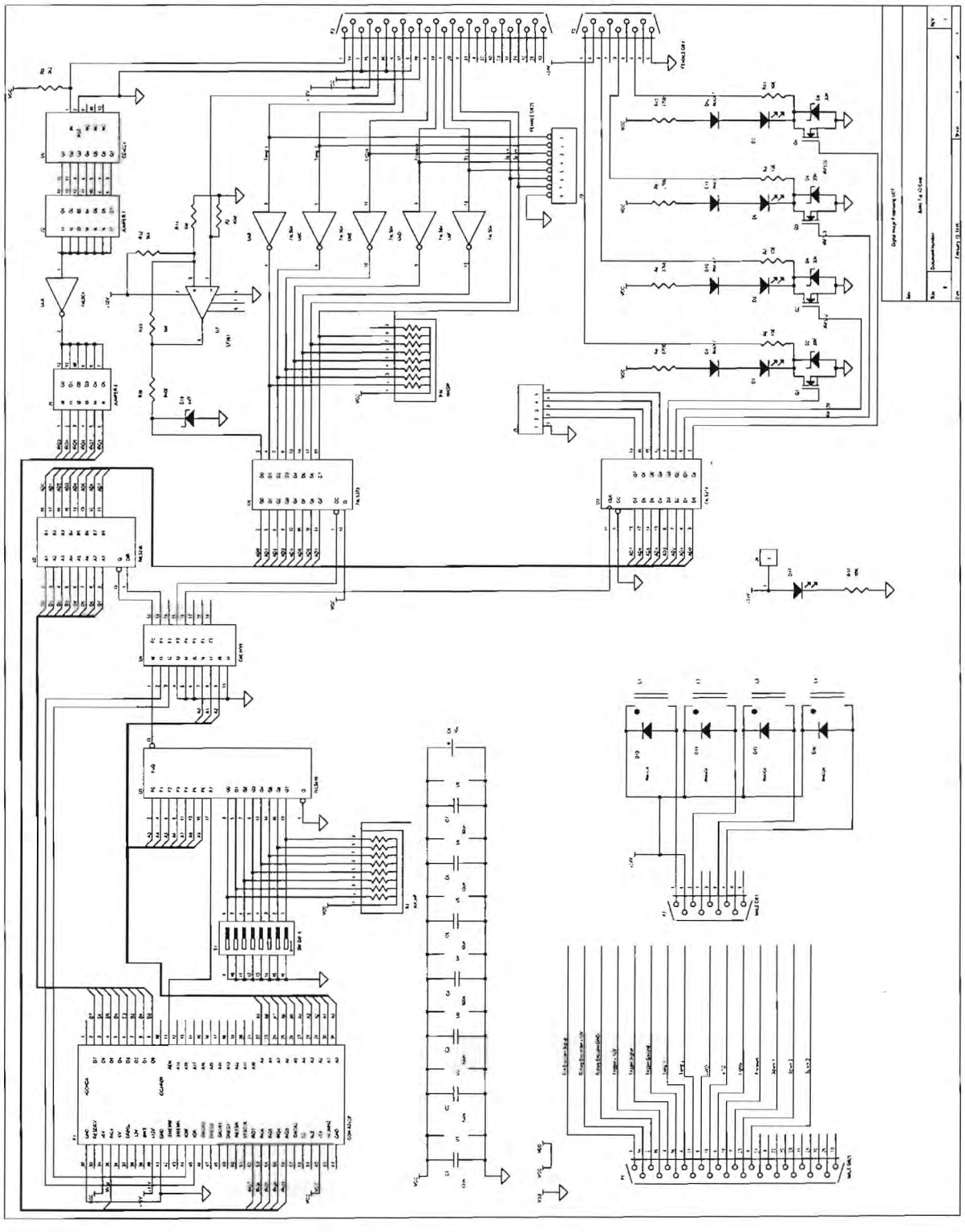
[24] Thierry Thomas and Michel Cattoen. Automatic inspection of simply patterned material in the textile industry. In Benjamin M. Dawson, Stephen S. Wilson, and Frederick Y. Wu, editors, *Machine Vision Applications in Industrial Inspection II*, volume 2183, pages 2–12. SPIE - The International Society for Optical Engineering, February 1994.

[25] M. Verman and S. Peleg. Min-max 0perators in texture analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-7:730–733, November 1985.

[26] Haruo Yoda, Yoza Ohuchi, Yuzi Taniguchi, and Masakazu Ejiri. An automated wafer inspection system using pipelined image processing techniques. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-10:4–16, January 1988.

# Appendix A

# Input/Output card schematic



# Appendix B

# Timing of closures in the star-wheel

The times in Table B.1 was determined by timing the rising edges of the detector pulse by polling the signal for a change. This was done for 1603 edges to get a good average. The last two columns in the table give the maximum and minimum times for each position on the star-wheel. The average time in for a closure to pass is $56.21ms$ when the lining machine is running at 1401 revolutions per minute.

| Slot number | Average time ($s$) | Minimum time ($s$) | Maximum time ($s$) |
|---|---|---|---|
| 1 | 0.055984 | 0.055197 | 0.056587 |
| 2 | 0.056313 | 0.055461 | 0.056866 |
| 3 | 0.055783 | 0.055138 | 0.056640 |
| 4 | 0.056287 | 0.055629 | 0.056907 |
| 5 | 0.056479 | 0.055847 | 0.057280 |
| 6 | 0.056229 | 0.055681 | 0.056888 |
| 7 | 0.056233 | 0.055686 | 0.057105 |
| 8 | 0.056190 | 0.055625 | 0.057028 |
| 9 | 0.056606 | 0.066799 | 0.057211 |
| 10 | 0.056025 | 0.055491 | 0.056853 |
| Average | 0.0562139 $\pm$ 0.000239 ($s$) | | |

Table B.1: Timing between slots on the star-wheel.

# Appendix C

# Data sheet for Pulnix $TM620$ asynchronous camera

# PULNiX TM-720/TM-620 ASYNC RESET HIGH RESOLUTION CCD CAMERA


## Features

- **Very high resolution 1/2" blemish free imager**
  **768(H) X 494(V) ... TM-720 (EIA)**
  **752(H) X 582(V) ... TM-620 (CCIR)**
- **Low light sensitivity (0.5 lux) with on-chip lenses**
- **Smearless shutter 1/60 to 1/10,000 (or 1/31,500)**
- **Asynchronous reset with ext. shutter control**
- **Ext. sync, interlace, non-interlace auto select**
- **Full frame integration**
- **Excellent S/N (50 dB)**
- **AGC on/off, gamma 1 or 0.45**
- **Small, light weight**

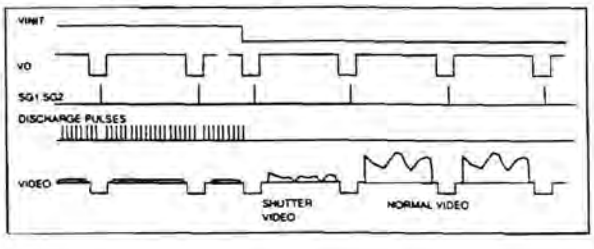
## General Description

The PULNiX TM-720 is a high resolution 765 H x 494 V black and white shutter camera with asynchronous reset capability and uniform MTF (Modulation Transfer Function) characteristics. These cameras are excellent in applications such as bar code reading, on-line inspection, gauging, character reading, high definition graphics, intensified CCD cameras, and detailed surveillance. Added to the wide versatility of this camera is an excellent low light sensitivity, despite the small size due to "on-chip" micro lenses.
Asynchronous reset and full frame integration are standard features of these cameras. AGC enable, internal IR cut filter, gamma adjust to 0.45, and the popular remoted imagers (standard 48") are all optional features that PULNiX offers for these cameras.

## Asynchronous Reset

The TM-720's asynchronous reset is flexible and takes external HD for phase locking. When VINIT pulse is applied, it resets the camera's scanning and purging of the CCD. There are three modes to control the asynchronous reset and shutter speed:

**1. External VINIT with double pulse** in which the duration of pulses controls the shutter speed externally.
**2. Internal shutter speed with Fast mode** in which the video signal has no delay from the reset timing (shutter speed range is 1/2000 to 1/31,000 sec.)
**3. Internal shutter speed with Slow mode** which can vary the speed control from 1/60 to 1/31,000 sec. The video signal starts with internal V reset timing related to shutter speed.

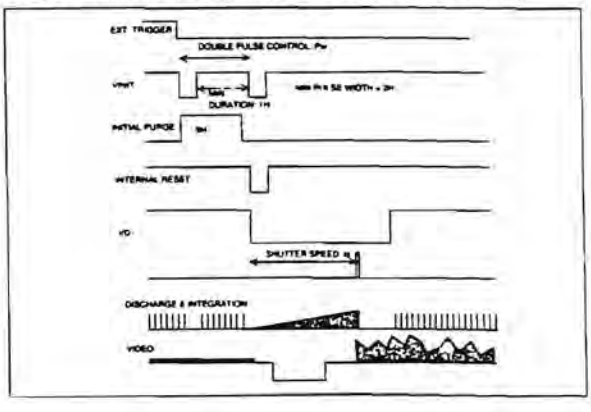



## Integration

The CCD imager of the TM-720 can be exposed longer than normal TV timing (16.7 msec.). This feature provides high sensitivity for dark environment applications. Integration is achieved by controlling the #11 pin of the 12-pin connector to Lo (GND). Because PULNiX uses an interline transfer chip in the TM-720, a full frame of resolution is obtainable with Frame mode option. (A full frame is not available in the shutter mode.) In integration, the signal process keeps optical black levels as reference black video to clamp video levels and this results in the cancelling out of thermal noise during the integration period.

## Electronic Shutter

The TM-720 has a substrate drain type shutter mechanism which provides a superb picture at various speeds without smearing.
Built in manual shutter speed control varies the electronic shutter rate between 1/60, 1/125, 1/250, 1/500, 1/1000, 1/2000, 1/4000, and 1/10,000 sec.
With VINIT high (5V), the CCD keeps discharging; with a negative going pulse to VINIT, the camera resets and purges the charge momentarily, then it starts integrating for the period of shutter control set by either an external double pulse or internal shutter control.



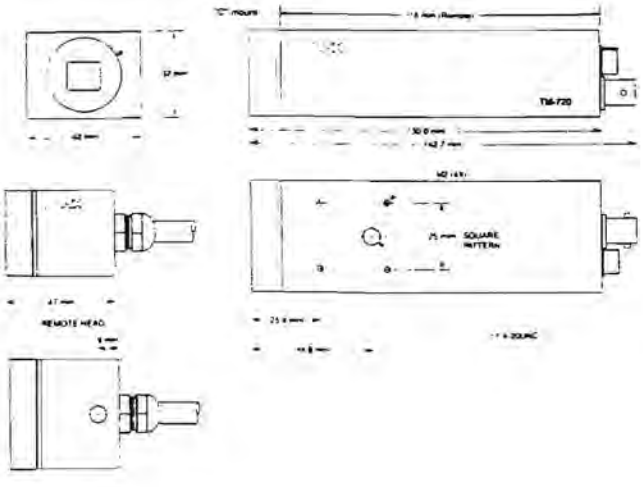
## Appendix C: Data sheet for Pulnix $TM620$ asynchronous camera

# Specifications

| Model | TM-720 (EIA) | TM-620 (CCIR) |
|---|---|---|
| Imager | 1/2" interline transfer CCD | |
| Pixel | 768 (H) X 494 (V) | 752 (H) X 582 (V) |
| Cell size | 8.4 µm X 9.8 µm | 8.4 µm X 8.2 µm |
| Scanning | 525 lines 60 Hz, 2:1 interlace | 625 lines 50Hz, 2:1 interlace |
| Sync | Internal/external auto switch<br>HD/VD, 4.0 Vp-p impedance 4.7KΩ<br>VD=interlace/non-interlace | |
| | HD=15.734kHz±5% | HD=15.625kHz±5% |
| H resolution | 570 (H) X 350 (V) | 560 (H) X 420 (V) |
| S/N ratio | 50dB min. | |
| Min. illumination | 0.5 lux f=1.4 without IR cut filter | |
| Video output | 1.0 Vp-p composite video, 75Ω | |
| AGC | ON/OFF (OFF std.) | |
| Gamma | 0.45 or 1.0 (1.0 std.) | |
| Lens mount | C-mount std. (shipped with CA-36 affixed), mini-bayonet optional | |
| Power req. | 12V DC, 250 mA | |
| Operating temp. | -10°C to 50°C | |
| Vibration & shock | Vibration: 7G (11Hz to 200 Hz) Shock: 70G | |
| Size (W X H X L) | 42mm X 32mm X 130mm<br>1.65" X 1.26" X 5.12" | |
| Weight | 190 grams | |
| Power cable | 12P-02 | |
| Power supply | K25-12V or PD-12 | |
| Auto iris connector | None | |
| Functional options | 1-1,2,3-1,4-1,10-1 to 10-6 (refer to video price list) | |
| Accessories | ENC-20 Video Detail Enhancer | |

Due to ongoing product improvements, specifications may change without notice.

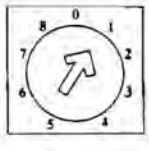
# Physical Dimensions




# Pin Configuration

**12-Pin Connector**

| | |
|---|---|
| 1 | GND |
| 2 | +12V |
| 3 | GND |
| 4 | Video |
| 5 | GND |
| 6 | VINIT |
| 7 | VD in |
| 8 | GND |
| 9 | HD in |
| 10 | GND |
| 11 | Int. cont. |
| 12 | GND |




**Manual Shutter Control**

| | |
|---|---|
| 0 | 1/60 |
| 1 | 1/125 |
| 2 | 1/250 |
| 3 | 1/500 |
| 4 | 1/1000 |
| 5 | 1/2000 |
| 6 | 1/4000 |
| 7 | 1/10,000 |
| 8 | NC |


In the U.S.
PULNiX America Inc.
770 Lucerne Drive
Sunnyvale, CA 94086
Tel: 408-733-1560
Tel: 800-445-5444
Fax: 408-737-2966

In the U.K.
PULNiX Europe Ltd.
Awary Court, Wade Road
Basingstoke, Hants RG24 OPL
Tel: 0256-475555
Tlx: 859532
Fax: 0256-466268

PULNiX
Industrial Products Division

In Germany
PULNiX Europe Ltd.
Sternbruch 5
D-8755 Alzenau
Tel: 06023-4666
Fax: 06023-4667

In Japan
Takanaka System Co. Ltd.
7-1, Shinomiya Narano Cho
Yamashina-ku, Kyoto, 607
Tel 075-593-9787