AUTOMATED VISION-BASED QUALITY INSPECTION OF BOTTLES FOR THE WATER BOTTLING INDUSTRY

A Thesis submitted in partial fulfillment of the

requirements for the degree

of

B.TECH MECHANICAL ENGINEERING

by

MITHUN GEORGE JACOB
03BME065

SCHOOL OF MECHANICAL AND BUILDING SCIENCES

VIT UNIVERSITY
(Estd U/s 3 of UGC Act 1956)
Vellore - 632014, Tamil Nadu. India
www. vit.ac.in

MAY, 2007

بسم الله الرحمن الرحيم

☎: 2457340
Fax.: (965) 2457766
C. R.: 30366

AlRawdatain Water Bott Co S.A.K. Closed

شركة تعبئة مياه الروضتين ش.م.ك مقفلة

☎ : ٢٤٥٧٣٤٠
فاكسميلي: ٢٤٥٧٧٦٦ (٩٦٥)
س.ت: ٣٠٣٦٦

Ref: FC212/07 الأشارة:

Date: 12/03/2007 التاريخ:

PROJECT COMPLETION CERTIFICATE

This is to state that **Mr. Mithun George Jacob (03BME065)** doing **B. Tech Mechanical Engineering** at the **Vellore Institute of Technology, India** has successfully completed his project work in our plant under the guidance of our Chief Maintenance Engineer, **Mr. John Thomas**.

The period for this work was from **November 29th 2006** to **March 11th 2007**. The project which was successfully incorporated into two of our production lines consisted of detecting and alerting the operators in time to rectify quality defects. The detected defects are missing bottles, missing and askew caps in the packaging which is a cardboard tray shrink wrapped in polyethylene at the next station.

We are pleased to say that we are extremely happy with the outcome which has completely resolved the problem of missing caps and bottles. This has been proved by studies conducted by our Quality Control Team and the feedback from our product distributors.

We have no hesitation to state that Mr. Mithun Jacob has implemented a Quality Control system on two of our lines which was completely successful, elegant and at the same time cost us next to nothing.

12/3/07.

Jacob George
(Factory Manager)

إدارة الإنتاج
رئيس قسم الإنتاج

12/3/07

Sunny Kuriakose
(Chief of Quality Control)

TÜV CERT ISO 9001: 2000
الشركة حاصلة على شهادة الأيزو ٩٠٠١

TÜV CERT ISO 14001 : 2004
الشركة حاصلة على شهادة الأيزو ١٤٠٠١

TÜV
HACCP CERTIFIED

ACKNOWLEDGEMENT

I extend my sincere thanks to the respected Chancellor **Mr. G. Vishwanathan**, the Pro-Chancellor (Academic), **Mr. G. Shankar Vishwanathan**, the Pro-Chancellor (Administration), **Mr. G. Sampath**, and the Vice Chancellor **Dr. P. Radhakrishnan** for providing the necessary facilities to complete my project successfully.

I wish to express my sincerest gratitude and thanks to **Dr. S. Narayanan**, Dean, School of Mechanical and Building Sciences for his complete and cooperative support without which this project would never have been completed.

My sincere thanks go to my internal guide, **Mr. V. G. Sridhar**, Selection-Grade Lecturer for his guidance, valuable suggestions and motivation during the course of my project.

I am deeply indebted to my external guide **Mr. John Thomas**, Chief Maintenance Engineer for his support at Al Rawdatain Water Bottling Co. (SAK Closed) as well as **Mr. Jacob George**, Factory Manager without whose technical expertise and pragmatic work ethic, this project would never have been conceived, completed or utilized.

I convey my sincerest thanks to my Project Coordinator, **Mr. Annamalai** for his unfailing supervision, and advice throughout the duration project.

ABSTRACT

An automated visual inspection (AVI) system based on computer vision has been developed to ensure quality control in the packaging of PET water bottles. This system utilizes low-cost webcam for image acquisition, a PC for image processing and a set of custom made algorithms. The algorithms have been written to compensate any deficiencies of lighting through preprocessing, and to rapidly segment and perform feature extraction on the image. Once the defects (missing bottles and missing or askew caps) have been identified, the system generates an alarm which alerts the machine operators immediately allowing them to take corrective action to either replace the tray or fix the defect.

Since the developed system is computer vision-based, it boasts a user-friendly GUI interface which allows the company to easily calibrate the system during installation or maintenance of the system.

Another advantage of using a PC-based system lies in the flexibility it offers. The system can be easily extended to solve similar problems such as moving tray-stationary camera or stationary tray-moving camera with equal ease. PC-based systems also allow the company to include extra features they might be interested like production data logging and incorporating traceability in the form of storage of images or maintaining a log file.

CONTENTS

Page No.

LIST OF TABLES

LIST OF FIGURES

NOMENCLATURE

f_c	-	Focal Length of the camera
S_c	-	Sensor Size of the camera
FOV	-	Area under inspection that the camera can acquire
WD	-	Distance from the front of the lens to the object under inspection
R,G,B	-	Red, Green and Blue component of a pixel
H,S,V	-	Hue, Saturation and Value component of a pixel
C'	-	Color Component (R,G,B,H,S,V) in unit form (C/255)
Dc	-	Chromatic Data of the Color Component
Δ	-	Delta RGB Value
d(x,y)	-	RGB Chromatic Data Function
f(m,n)	-	Value of the pixel located in the m^{th} column, n^{th} row
G	-	Pixel value after thresholding according to condition *i*
T_i	-	Threshold Value *i*
I(i,j)	-	Value of pixel in the image located in the i^{th} column and n^{th} row
K(k,l)	-	Value of kernel element located in the k^{th} column and l^{th} row
O(i,j)	-	Value of *I(i,j)* after convolution
w_c	-	Value assigned to represent an edge
$O_{x,y}$	-	Edge Point at I(x,y)
$O_{a,b}$	-	A random point selected from the stack of edge points
D(O,O')	-	Euclidean Distance between two points O and O'
Dmin	-	Minimum allowable distance between two edge points of the same BLOB

AVI	-	Automated Visual Inspection
MER	-	Minimum Enclosing Rectangle
BLOB	-	Binary Large OBject

CHAPTER 1

INTRODUCTION

1.1 Introductory Remarks / Outline

Vision based solutions offer several advantages over conventional quality inspection systems due to their flexibility and convertibility and is generally a low cost solution. Several applications of machine vision have been discovered and utilized such as quality inspection, object recognition and object tracking. The solution discussed here concerns itself with quality inspection.

This chapter deals with the immense of application of automated visual inspection systems for achieving Total Quality Management (TQM), packaging automation and also the advantages of computer vision-based automation followed by a brief description of the system architecture.

1.2 Zero Defects and Automated Visual Inspection Systems

'Zero Defects', implemented by preventing established and known problems is an essential constituent of total quality management. Inspection is no longer the backbone of quality, but it has an important role in preventing and identifying those problems, and in limiting their impact on the company and, even more importantly, on its customers.

Automated visual inspection (AVI) systems can do this inspection by simulating human vision, without the wastage of human intelligence to look at the results. Although no AVI today could ever attain the levels of versatility, flexibility and discrimination of human vision, these systems compensate these demerits with their ability to work continuously, not get tired, and maintain a constant level of accuracy as well as working faster than any human inspector.

Therefore, AVI can be utilized as a powerful tool for monitoring and hence improving the quality of operations within the industry. But the issue which needs to be addressed is how AVI can be harnessed to work towards zero-defect manufacturing.

1.2.1 Inspection Function

There are three types of inspection functions [1-7]:

1.2.1.1 Judgment Inspection

Judgment Inspection concerns itself with merely discovering defects in the product. This characteristic, albeit useful, makes no attempt to feedback its information to help to improve either product of processes. It is therefore of little use in improving quality of operations.

1.2.1.2 Informative Inspection

Informative Inspection concerns itself with merely reducing defects in the product. This characteristic involves feeding back information to the process concerned when a defect has been identified. Informative inspection can be used for statistical process control, successive check systems or self check systems.

1.2.1.3 Source Inspection

Source Inspection aims to eliminate defects in the product at the source itself. It is based on the idea of discovering errors in conditions that give rise to defects and performing feedback and action at the error stage so as to keep those errors from turning into defects [1].

1.2.2 AVI: Inspection, QC, QA and TQM

Input inspection systems can reduce costs, by preventing defective material from going through any processing and protection against incompetent suppliers.

However, isolated, they cannot aid the organization in quality improvement. Because the company cannot directly control the quality of its supplies, this AV1 is serving the traditional function of inspection. In-process and output inspections can be used either online or offline. When used offline, they will improve quality by the information they feed back to the controlled process. Since inspection is carried out away from the processes themselves, offline systems arc detection-based: AV1 then becomes a quality control tool.

Online post-process systems will improve quality by adjusting the quality output from the controlled process and thus eliminating the possibility of defects reaching the next process (or the customer). These systems are reactive, rather than merely preventive, to defects, and will act on the process only if it deviates from proper operation. This sort of AV1 is a quality assurance tool.

Finally, online in-process systems are purely preventive, aiming to eliminate defects by correcting mistakes at the error stage, before they become defects. AV1 in this case is a true TQM tool.

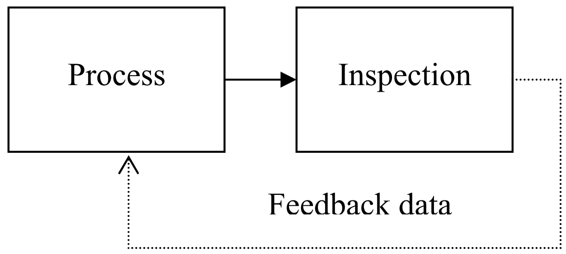


Fig. 1.1 AVI Online In-process Inspection

1.3 Computer Vision-based Automation

The appropriate approach to take in implementing a machine vision system has been a point of argument time and time again. Machine Vision Systems can be broadly classified under 3 categories:

1.3.1 PC-based Automation

A PC-based AVI implies the use of an implementation based on a PC. This could be either using a camera with the capability to interface directly to the PC (IEEE 1394/Firewire, CameraLink, LVDS, USB, etc.) or a system based on a frame grabber or some other intelligent image processing board or vision engine that plugs into the PC.

1.3.2 Smart Cameras

Smart Cameras are self-contained units which include the image sensor as well as the "intelligence" and related I/O capabilities. Most vision sensors have a limited and fixed performance envelope whereas Smart Cameras possess more flexible tools and are inherently capable of being programmed to handle a plethora of imaging algorithms and application functions.

A point to be noted is that a PC-based vision system is generally recognized as having the greatest flexibility and, therefore, capable of handling a wider range of applications. One significant difference is that vision sensors/Smart Cameras are essentially single socket units, while PC-based vision systems can generally handle multiple camera inputs.

1.3.3 Hybrid Systems

Hybrid Systems fall somewhere between the PC-based vision system and a Smart Camera/vision sensor is what some call an "embedded vision computer." This type system is essentially a stand-alone box with frame storage and intelligence. It generally has limited flexibility and comes with a number of fixed application-specific routines. These are distinct from Smart Cameras in that the camera is tethered to the unit rather than self-contained. They often have the ability to handle multiple camera arrangements, which can be useful for many applications.

In the next section, we will look at a set of arguments debating the use of a PC-based vision system or a Smart Camera for this particular machine vision application.

1.3.4 PC-based Vision system

1.3.4.1 Development

When developing a machine vision solution to an industrial environment, the system integrator usually does not exactly know where the problem analysis will actually take him. It is not unusual that the requirements for processing power or functional capabilities turn out to be poorly foreseen, and the final solution is never simpler than the initial idea.

The PC-based solution offers an immense set of potential resources, in terms of computational or interfacing performance. The PC platform is essentially open, and it became so popular that its cost to performance ratio is unbeatable.

1.3.4.2 Cost-effective

The low-cost argument is particularly true for the desktop PC, but it is sometimes claimed that the mechanical weakness of a mainstream desktop PC is not compatible with the industrial requirements of a serious machine vision application. However, compared to the more expensive industrial PCs, the low-cost desktop PC offers the latest CPUs and associated components, offering the highest performance at the lowest cost.

All in all, when considering all trade-offs to be made in its design, the machine vision developer reaches the conclusion that the PC-based system is the most cost-effective solution. It best suits his need for functional evolution during the design stage, and even after when upgrading the system becomes an issue. Upgrading software is an easy way to improve functionality, and upgrading the PC hardware is an easy to improve the performance.

1.3.4.3 Computational Complexity

PC-based machine vision systems are generally more capable than Smart Camera based systems. They have more computational horsepower to be able to handle much more sophisticated software algorithms. The Smart Cameras are great for simple tasks using general edge detection or binary tools; however, they do not have the computational power or memory to handle more sophisticated application specific algorithms. They will be limited to how fast and how complex the inspection performed will be.

1.3.4.4 Flexibility and Power

The PC offers greater flexibility in the number of options that can be selected. For example one can use a line scan versus an area scan camera with the PC. One can use third party software packages with the PC approach (Smart Cameras tend to be single source software).

PC's tend to offer greater power and speed due in large part to the speed of the Intel processors used internally. This power in turn means that PC's are used to handle the "tougher" applications in machine vision.

1.3.4.5 Ideal for Robot Applications

Smart Camera based vision systems are rudimentary when it comes to calibration as the robot or mechanism has to be calibrated separately. Typical PC based vision systems can handle up to four cameras per frame grabber. In applications that require multiple cameras, the cost of a PC based system should be compared with the cost of multiple Smart Cameras. In order to use Smart Cameras the customer has to select and link separate low-cost products. With standardized controller-based vision system the customer is investing in a pre-engineered and pre-configured system.

1.4 Packaging Automation

AVI has been implemented successfully for the inspection of a variety of packaging situations such as the inspection of IC Pads and Bonds [8], quality grading of herring roe within a high-volume extraction-processing-packaging line [9], the production control and automatic packaging of plastic air sleeve guides [10], automatic inspection of the connecting part of the wire bond of an IC [11], and many other applications. Some of the common hurdles to be overcome whilst setting up a vision-based solution for an industrial scenario have been delineated below:

1.4.1 Lighting

Formerly, “staging” an application—that is, physically setting up the lighting, camera and part to be inspected, and divining the correct parameters for all three—was generally a difficult and time-consuming job. And once the system was in operation, changes in ambient lighting or atmospheric conditions within the plant often threw the carefully calibrated lighting for a loop.

Increases in computing power have allowed vision system providers to employ software algorithms that are better able to handle variations in lighting. Lighting components exhibit greater stability and can be customized to the application.

1.4.2 Traceability

Vision sensors are invaluable in the terms of traceability. In today’s world, traceability is of paramount importance as it allows the overseers to track and trace, either through an image that can be captured and archived, or through a bar code. In addition to reading bar code, though, a vision system can perform quality and product safety operations. With the aid of vision sensors, it is possible to tell whether a safety ring is on, or on correctly, prove that the correct label is put on the correct container, check for foreign matter in liquids by verifying fill level since foreign matter will raise the level. It can also perform sorting and grading operations that can help improve production. Such things cannot be achieved with a simple laser bar code reader. Even in terms of bar code reading, 2D bar codes can be read easily by a vision system, while it can be expensive to configure a laser bar code reader to perform the same task [12].

The need for traceability has become paramount for US approval after the issuance of the Food and Drug Administration's (FDA) 2001 allergens regulations and the Bioterrorism Act of 2002. The FDA allergens regulations require that food that comes into contact with any of eight common allergens, which together cause approximately 90 percent of allergic reactions in the United States each year, be clearly labeled. The Bioterrorism Act stipulates what's been called "one step back, one step forward," that is, each part of the food chain must be able to trace its source back one step, and trace the destination of the food one step forward.

1.4.3 Effects of Product Development for Consumers

Digital imaging technology spawned by the frenetic pace of product development in the consumer realm has impacted vision systems. This is evident from the plethora of smaller cameras with excellent resolution and a smaller price tag.

Vision applications have achieved a significant role in the packaging industry because vision is a much more efficient way of checking products and also because vision systems are getting easier to use. This ease of use comes primarily from the software that are available now. Developing vision systems have become much easier with the aid of a large number of highly efficient function libraries and graphical programming environments which have drastically simplified many programming tasks.

1.5 Problem Definition

The process object of the automation involves the quality inspection of the finished product (a tray of PET water bottles) which was manned by human labor. Due to large production and high speed operations, a quality inspection method based on human labor is prone to error due to fatigue and boredom. An attempt is made to implement an AVI as an alternate quality inspection method.

Due to the nature of the machine and the errors committed by human labour, the defects passed by will eventually become a serious quality issue once the customer finds a tray with partially missing contents. (Fig. 1.2.).

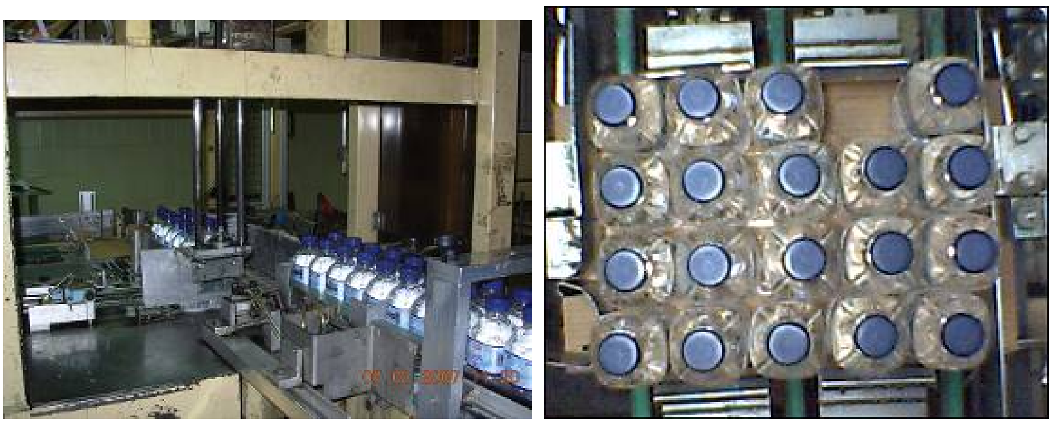


Fig.1.2: 0.5L Packing Line and Sample Defective Tray (Top View)

This solution is designed to efficiently detect the absence of a bottle in the carton in short time duration and take effective measures to discard the defective product.

Several industries employ the use of photoelectric sensors, load cells or dual fiber-optic sensors [13] to count and inspect bottles exiting the production line. The disadvantage of using such methods is the high cost incurred and the lack of flexibility.

Hence, a vision-based solution is designed for increased flexibility as well as low cost. It employs an image acquisition device, a PC, and a set of image processing algorithms to process the acquired image. The challenge of designing a vision-based solution for this application lies in the high amounts of specular reflection present in the image due to water present in the bottle which hinders effective segmentation. Therefore, this calls for a novel approach to be used in this solution.

AVI must be installed on two separate production lines; the first production line (PL1) involves the tray remaining stationary in the assembly line for minimum time duration of 6 seconds and the second production line (PL2) involves a constantly moving tray. Images describing similar aspects of both lines will be used in this thesis.

1.6 Scene Constraints

One of the primary hurdles facing the development of any AVI is the seamless integration of the AVI into the production line. In order to achieve this, the image acquisition device must be installed satisfying all the required criteria (vibration free, adjustable to some extent, etc.) as well as not interfering with the machine operation in any way. The construction of the camera mount (Fig. 1.3) will be described in forthcoming chapters.

Another crucial factor which must be accounted for is the installation of adequate lighting for the efficient acquisition of images. It has been found that the best lighting for the processing of reflective, specular surfaces is diffused and domed lighting [14]. This has been implemented by installing a fluorescent bulb with a reflective dome mounted behind the bulb.

Fig 1.3. Diffused Domed Lighting and the Camera Mount from PL2

The proposed AVI has been implemented in an industrial scenario where limitations were imposed on the position of the lighting resulting in its unsatisfactory position as observed in Fig. 1.3. This disadvantage has been rectified with the aid of a set of image processing algorithms which pre-process the image so that the pre-processed image can be easily used for segmentation without any loss in data.

1.7 Image Acquisition

The image acquisition subsystem is implemented with standard components on a low-cost personal computer. The acquisition device used in this system is a webcam (1/3” CMOS sensor) capable of capturing images with bearable quality and is connected to the computer via the USB port.

1.7.1 CCD/Frame Grabber

A charge-coupled device (CCD) is an image sensor, consisting of an integrated circuit containing an array of linked, or coupled, light-sensitive capacitors. This device is also known as a Color-Capture Device.

CCDs containing grids of pixels are used in digital cameras, optical scanners and video cameras as light-sensing devices. They commonly respond to 70% of the incident light (meaning a quantum efficiency of about 70%) making them more efficient than photographic film, which captures only about 2% of the incident light.

1.7.1.1 Working

An image is projected by a lens on the capacitor array, causing each capacitor to accumulate an electric charge proportional to the light intensity at that location. A one-dimensional array, used in line-scan cameras, captures a single slice of the image, while a two-dimensional array, used in video and still cameras, captures the whole image or a rectangular portion of it. Once the array has been exposed to the image, a control circuit causes each capacitor to transfer its contents to its neighbor. The last capacitor in the array dumps its charge into an amplifier that converts the charge into a voltage. By repeating this process, the control circuit converts the entire contents of the array to a varying voltage, which it samples, digitizes and stores in memory. Stored images can be transferred to a printer, storage device or video display. CCDs are also widely used as sensors for astronomical telescopes, and night vision devices.

1.7.2 Webcam

A web camera (or webcam) is a real-time camera (usually, though not always, a video camera) whose images can be accessed using the World Wide Web, instant messaging, or a PC video calling application.

Web-accessible cameras typically involve a digital camera which uploads images to a web server, either continuously or at regular intervals. This may be achieved by a camera attached to a PC, or by dedicated hardware. Videoconferencing cameras typically take the form of a small camera connected directly to a PC. Analog cameras are also sometimes used (often of the sort used for closed-circuit television), connected to a video capture card and then directly or indirectly to the internet.

1.7.2.1 Working

Webcams typically include a lens, an image sensor, and some support electronics. Various lenses are available, the most common being a plastic lens that can be screwed in and out to set the camera's focus. Image sensors can be CMOS or CCD, the former being dominant for low-cost cameras, but CCD cameras do not necessarily outperform CMOS-based cameras in the low cost price range. Consumer webcams usually offer a resolution in the VGA region, at a rate of around 25 frames per second. The higher resolution of 1.3 Megapixel is also available from the brands Microsoft, Kinamax, Sabrent, Logitech, Vije, and HP.

Support electronics is present to read the image from the sensor and transmit it to the host computer. The camera (Fig.1.4), for example, uses a Sonix SN9C101 to transmit its image over USB. Some cameras - such as mobile phone cameras - use a CMOS sensor with supporting electronics 'on die', i.e. the sensor and the support electronics built on a single silicon chip, to save space and manufacturing costs.

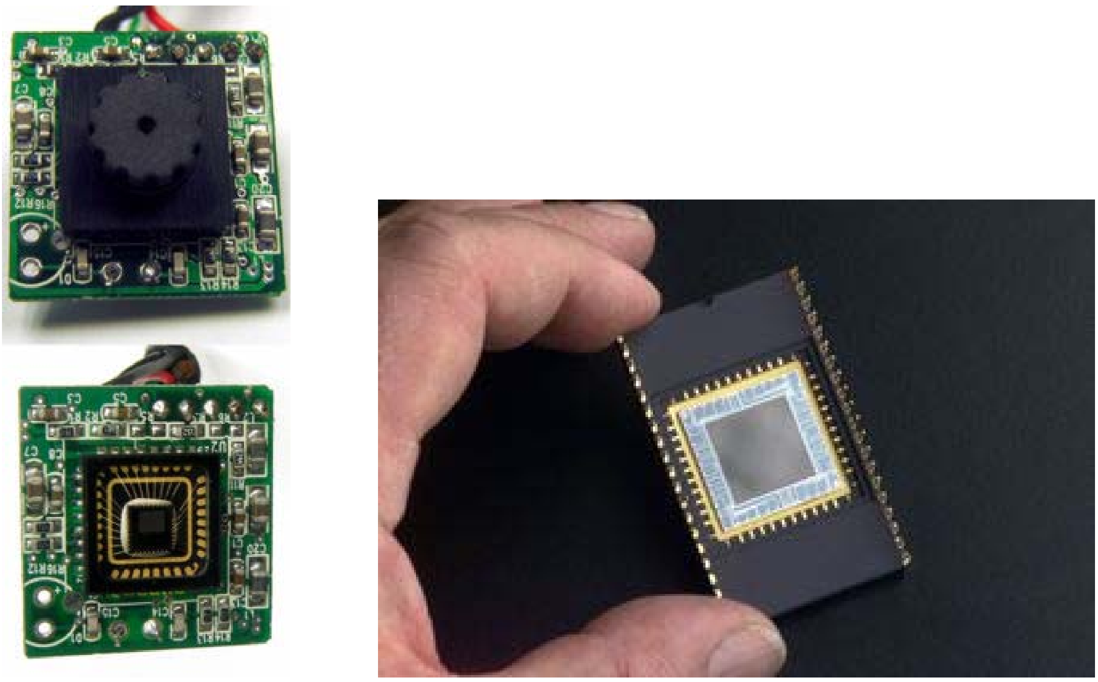


Fig 1.4. Webcam and CCD Circuitry

1.7.3 Application

The decision to utilize webcams with CMOS sensors as opposed to CCD sensors[15] is delineated in the following table comparing both sensors.

Criteria	**CCD**	**CMOS**	**Definition**	**Applies?**
Responsivity	×	√	Measure of signal level per unit of optical energy	**No**
Dynamic Range	√	×	Ratio of a pixel's saturation level to its signal threshold	**No**
Uniformity	√	×	Consistency of response for different pixels under identical illumination conditions	**Yes**
Speed	×	√	Speed of operation which depends on how distance should be communicated over and number of chips information must be piped through	**Yes**
Reliability	×	√	Reliability which depends on the level of integration on the chip and other factors	**Yes**
Cost	×	√	Cost of the sensor in general	**Yes**

Table 1.1. CCD vs CMOS

As observed from Table 1.1., it is apparent that a CMOS sensor would be perfect for this application for its superior speed, reliability and cost. The usage of a webcam with a CMOS sensor has proved to be quite adequate and has made it possible to implement this system at a low cost.

CHAPTER 2

LITERATURE REVIEW

The literature review conducted in this study is described below in the following sections:

2.1 Zero Defects and TQM

Panayiotis Panayiotou and Keith Ridgeway state that automated vision systems have become a powerful tool for quality control but they can also be used directly for quality improvement. They have stated that through the aid input inspection systems, offline inspection systems, online post-process systems and online in-process systems ensures that the AVI becomes a tool for inspection, quality control, quality assurance, and ultimately a true TQM tool respectively [7].

Omron Electronics LLV state that food & beverage companies invest heavily and consistently in solutions to optimize their manufacturing operations in order to avoid capital equipment investments to increase capacity. Slim profit margins compared to other industries, and increasing quality and variety demands from customers, retailers and regulatory agencies are typical reasons for taking this incremental improvement approach. They believe that in many cases, machine vision systems dramatically increase the number of inspections; reduce human error, eyestrain or repetitive motion injuries; and allow an increase in production speed and accuracy.

Waste prevention scenarios use vision as a diagnostic tool to detect which problem(s) to repair on an expensive piece of machinery. They also use vision to limit waste in product, packaging and marking materials when fill level is too high or too low; when the label is crooked; or when an ink jet marking system produces blobs instead of characters and to remove non-saleable product from the supply chain.

Even a slight downward readjustment of the waste threshold can generate sufficient savings to justify the cost of a vision inspection system and boost bottom line profitability. [13]

2.2 Handling Scene Constraints

Melles-Griot states that the basic approach to **lighting** for a particular application is easily determined as it is a function of the type of object and the features to be measured. For specular surfaces such as ones encountered in filled water bottles, diffuse front illumination has been found to be optimal as it is soft, relatively non-directional and also reduces glare on specular surfaces. Coupled with the fact that it is relatively easy to implement, and that once the lighting is domed, it eliminates glares and shadows as well.

Diffused and domed lighting can be obtained from easily-available fluorescent lamps (both straight tube and ring lights) as they are inherently diffuse. Diffuse illumination of specular surfaces allows imaging without bright reflections. Surface texture is minimized, and there is less sensitivity to surface angles on parts. These lamps are highly desirable as wide or narrow spectral ranges available and are efficient and long lived. [14]

Malamas, Petrakis, Zervakis, Petit, and Legat state that CMOS sensors have distinct advantages over CCD sensors in the terms of responsibity, speed, reliability and cost. They believe that CCDs offer superior image performance and flexibility in terms of system size but CMOS imagers offer more integration, lower power dissipation and smaller system sizes at the expense of image quality and flexibility. But problems such as the low-quality issue typically associated with CMOS sensors have been tackled by the use of microlenses.

Microlenses are small lenses manufactured directly above the pixel to focus the light towards the active portion and the minimization of the space circuitry in the CMOS pixel. [15]

2.3 Applications of AVI

Sreenivasan, Srinath, and Khotanzad state that one of the problems in increasing reliability in the manufacture of integrated circuit devices is inspection of the bond pads and the bonds connecting the bond pads to the lead fingers of the device. The continuing increase in packing density of VLSI circuits requires that the inspection process be completely automated.

Their work presents methods for visual inspection of bond pads and bonds, which are intended to automatically extract parameters of significance in determining their quality, from two-dimensional images taken from the top of the IC wafer. [8]

Beatty, Gosine, and C.W. de Silva presents ongoing research in knowledge based vision which addresses the problem of quality grading within the fish processing industry. In particular, the problem of grading herring roe within a high-volume extraction-processing-packaging line is considered in the light of high-speed computer vision and pattern recognition techniques that form the basis for a ’grading expert’ that automatically grades individual pieces of roe according to an existing grading scheme. [9]

García, Incertis, Trespaderne, López, and González describes the functioning of a manufacturing cell, in which several complex systems have been joined, such as: a robot, a computer vision system, and a lathe. To allow communication between these systems in an organised manner, diverse technologies have been used, varying from a field bus to the intranet of the company.

The robot takes the role of master, coordinating the functioning of the rest of subsystems. The computer vision takes charge calculating the position and orientation of the pieces of injected plastic and communicating them to the robot. The robot gathers the pieces and takes them up to the lathe. Once the process of the lathe has finished, the robot gathers again the piece and deposits it in the packaging system. The cell is endowed with the possibility of remote adjustment and monitoring across the company's Intranet. The cell's functioning conditions can be modified from the laboratory, to assure that the pieces made are within the quality limits specified by the clients. [10]

Khotanzad, Bannerjee, and Srinath paper describes a vision system for automatic inspection of the connecting part of the wire bond of an IC where the wire connects, to the bond pad on the chip. It considers a popular type of such bonds known as "ball bond". Using two-dimensional images taken from the top of the IC wafer, the system determines several geometric measures which are important in determining the quality of the bond.

These measures include the boundary, length major and minor axes of the best fitting ellipse and the center. The process utilizes automatic thresholding, morphological operations and geometric moments of the image. The success of their method has been demonstrated through experimental studies on actual bonds. [11]

Xiangju, Guoliang, and Yunkuan state that computer vision and pattern recognition technologies make robots' eyes possible, which has rapidly promoted the development of robotic sciences. Barcode, which is simple but contains much information, has been used in navigation, manufacture and other situations by robots. Reading a barcode based on image analysis is an essential technique for such a robot system. Unfortunately, few existing image analysis methods about it are capable of identifying the codes under uncertain conditions, such as varying illumination, different sizes or orientations.

Their work presents a method which can locate barcode region at an uncertain 3-D background through hierarchically extracting the distinct features of it with a high precision even for the broken, aberrant, partly blotted out stripes and other poor qualities of images. [12]

Nunnally and Weiss have developed a robotic arm with two degrees of freedom for applications involving machine vision. The unique features of this system are its ability to perform simultaneous real-time image acquisition and range sensing; its flexible design, which confers the ability to handle a wide variety of tasks: and its low cost. These desirable features make the arm suitable for use in both industrial and educational environments.

They developed a simple, economical robotic arm with both image acquisition and range sensing capabilities for machine vision applications. It is connected via an interface board to an IBM PC/AT microcomputer, offering maximum flexibility in programming the arm for different tasks. A sonar ranging device and a video camera are mounted on the arm, and gives the system 3-D vision capabilities. The sonar ranging device and associated clock/counter circuitry interfaces to the computer via the same board as the robotic arm, whereas the video camera is connected to an image digitization and display board.

This system is economical and flexible enough for a wide range of applications. To illustrate its capabilities, the arm was programmed to scan the immediate environment and attempt to locate and identify the nearest object. Standard image processing and computer vision techniques such as binary thresholding, histogram analysis, signatures, and pattern recognition were applied, and excellent object recognition was achieved under a wide variety of lighting as well as distance conditions. [16]

Chivilo, Mezzaro, Sgorbissa, and Zaccaria present a robotic attendant, i.e. a mobile platform equipped with a TV camera which is capable of following a human leader along complex trajectories. Differently from existing systems which mainly rely on the detection of colour blobs or particular features/markers, our system is based on the detection of motion through optical flow computation, thus being implicitly able to follow both persons and other robots with different physical characteristics in terms of colour, shape, etc.[17]

Zhang, Koh, and Wong describes a low cost and high speed visual inspection system for the automatic measurement of wire-bond ball height using structured lighting technique.

They believe that this method is superior to other possible techniques reported so far because of its potential in on-line implementation. The system can measure the wire-bond height to an accuracy of +3 micron. Higher accuracy can be achieved with higher resolution camera and video frame grabber. [18]

Sari-Sarraf, and Goddard, Jr describes a vision-based fabric inspection system that accomplishes on-loom inspection of the fabric under construction with 100% coverage. The inspection system, which offers a scalable open architecture, can be manufactured at relatively low cost using off-the-shelf components.

While synchronized to the motion of the loom, the developed system first acquires very high-quality vibration-free images of the fabric using either front or backlighting. Then, the acquired images are subjected to a novel defect segmentation algorithm, which is based on the concepts of wavelet transform, image fusion, and the correlation dimension.

The efficacy of this algorithm, as well as the overall inspection system, has been tested thoroughly under realistic conditions. The system was used to acquire and to analyze more than 3700 images of fabrics that were constructed with two different types of yarn. In each case, the performance of the system was evaluated as an operator introduced defects from 26 categories into the weaving process. The overall detection rate of the presented approach was found to be 89% with a localization accuracy of 0.2 in (i.e., the minimum defect size) and a false alarm rate of 2.5%.

Their contribution is highly important as they have implemented a low-cost vision-based inspection system which is capable of running exceedingly effectively in real time. [19]

Strand, Lowry, Lu, Nelson, Nikkel, Pocha, and Young have successfully constructed a low-cost vision-based fiber pigtailing system as opposed to using highly expensive optoelectronic (OE) devices.

The high cost of OE devices is due mainly to the labor-intensive packaging process. Manually pigtailing such devices as single-mode laser diodes and modulators are very time-consuming with poor quality control. A fully automated system must include high-precision fiber alignment, fiber attachment techniques, in-situ quality control, and parts handling and feeding. [20]

CHAPTER 3

CONSTRUCTION

The construction of various elements in the system is crucial to its smooth functioning. The construction of central elements such as the camera mount has been described below followed by a discussion on the use of an image acquisition trigger for both production lines. The issue of lightning has already been discussed in the Chapter 1 (Section 1.6.).

3.1 Camera Mount

The camera mount is a vital element in the efficient running of the AVI. In order to implement seamless integration of the AVI into the production line without compromising on the required criteria such as it should be vibration free, should not interfere with production or maintenance in anyway and a certain degree of freedom.

3.1.1 Hardware Integration and Fabrication

In order to ensure that the camera is installed seamlessly without disrupting production in anyway be it installation or maintenance, the camera was fixed overhead the tray and bolted onto the frame above it.

The position of the camera was chosen with great care to ensure that no moving/stationary parts obstructed the field of view (FOV) in any way. Great care was taken to ensure that the camera was placed at an appropriate working distance so that the FOV would comfortably encompass the tray as well as the image acquisition trigger for PL1.

In the case of PL1, a redundant object was removed from the frame which could accommodate a custom-made mount for the camera (Fig. 3.1).

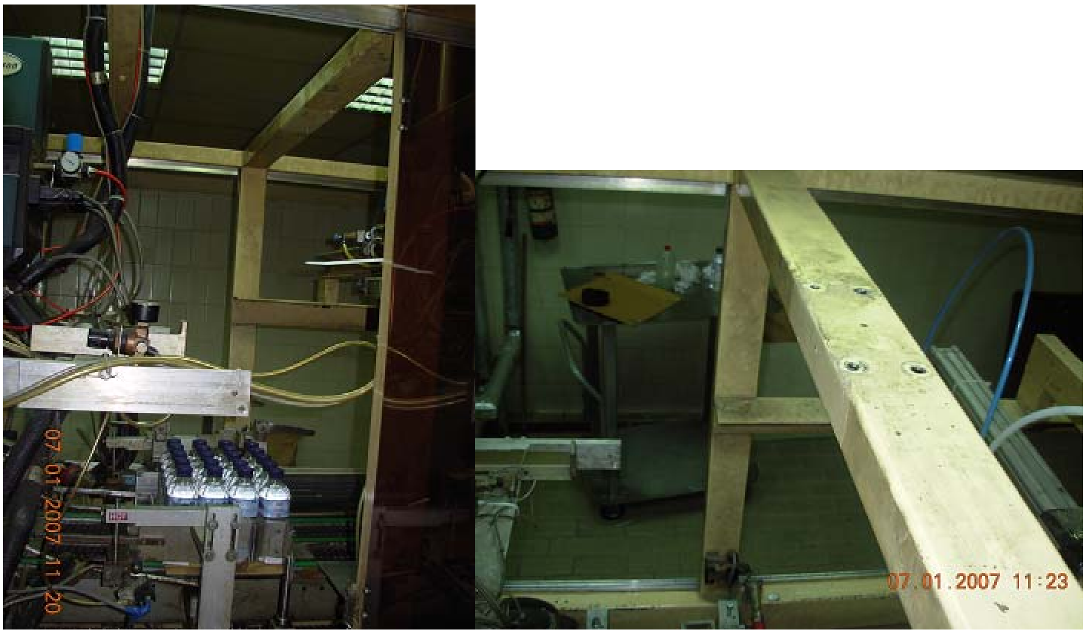


Fig. 3.1. Placement for Camera Mount

Once the dimensions of the holes had been taken, and the distance by which the camera should project from the frame so that the camera lens would be vertically above the center of the tray had been ascertained, the camera mount was fabricated.

In order to provide the mount with a certain degree of freedom, the following simple but effective extension was devised (Fig. 3.3). By adjusting the position of the nuts, it is possible to move the camera horizontally, and vertically as desired. It is also possible to change the angle of orientation of the lens by turning the bolt.

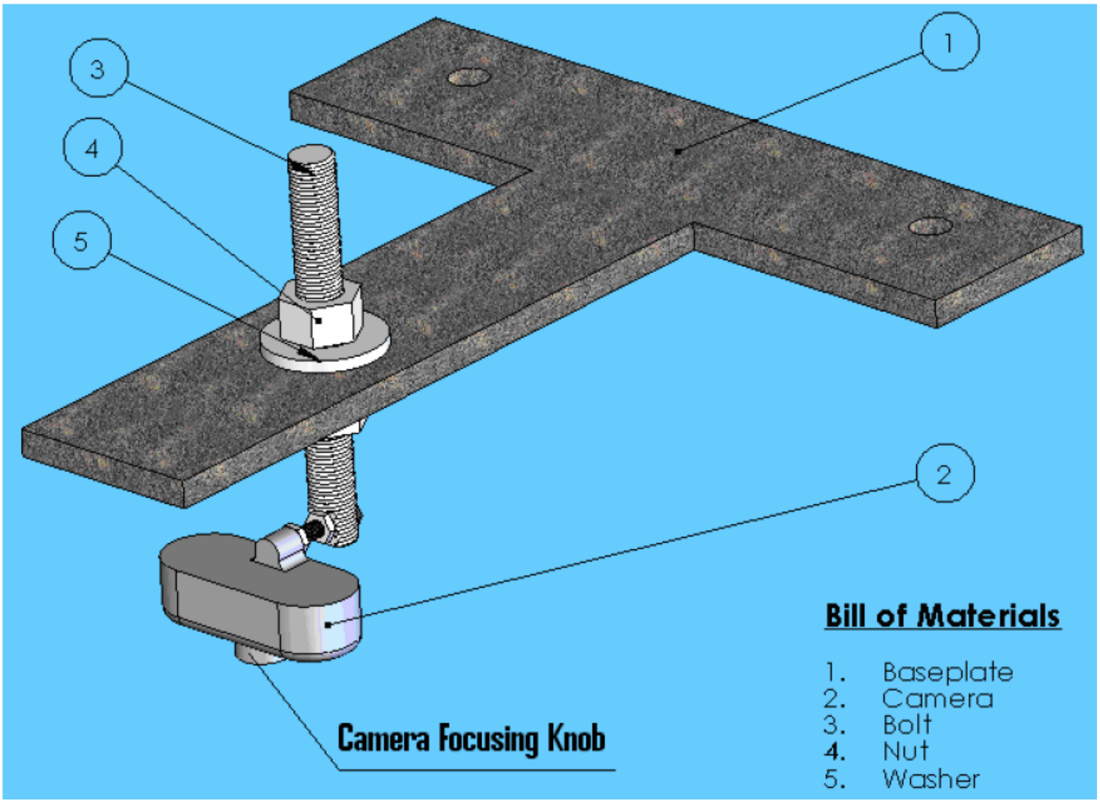


Fig. 3.2. Camera Mount (Model)

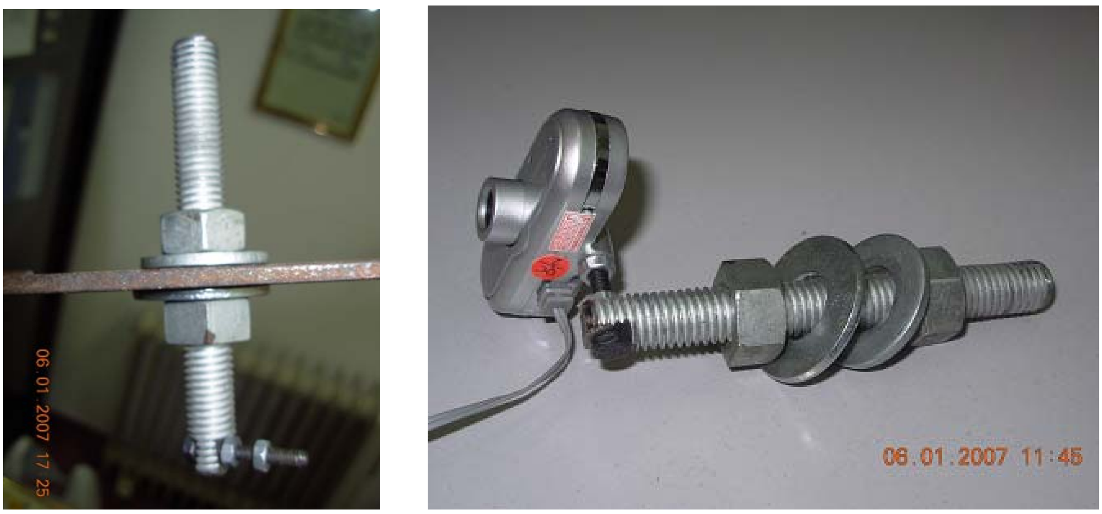


Fig 3.3 Adjustable Mount

Using the adjustable mount, the camera was mounted on the frame successfully and by accurately designing the dimensions of the baseplate, the camera lens was vertically over the center of the tray. Fig. 3.4 displays the installed camera mount.

Fig. 3.4. Installed Camera Mount for PL1

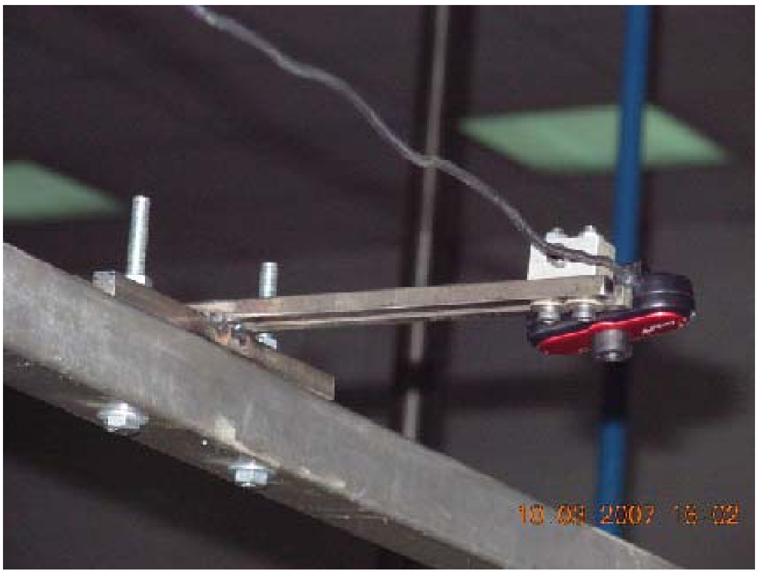


Fig. 3.5. Installed Camera Mount for PL2

Similarly, the calculations and fabrication was implemented for the camera mount for PL2. Since there were no redundant objects in the PL2 frame, two holes were drilled and the mount displayed in Fig. 3.5 was installed on the frame.

3.1.2 Field of View vs. Working Distance

The Field of View (FOV) is defined as the area under inspection that the camera can acquire and the Working Distance (WD) is defined as the distance from the front of the lens to the object under inspection.

It is vital that the FOV encompass the entire tray as well as the image acquisition trigger (in PL1) and hence, the minimum WD must be calculated in advance to facilitate seamless integration of the camera mount.

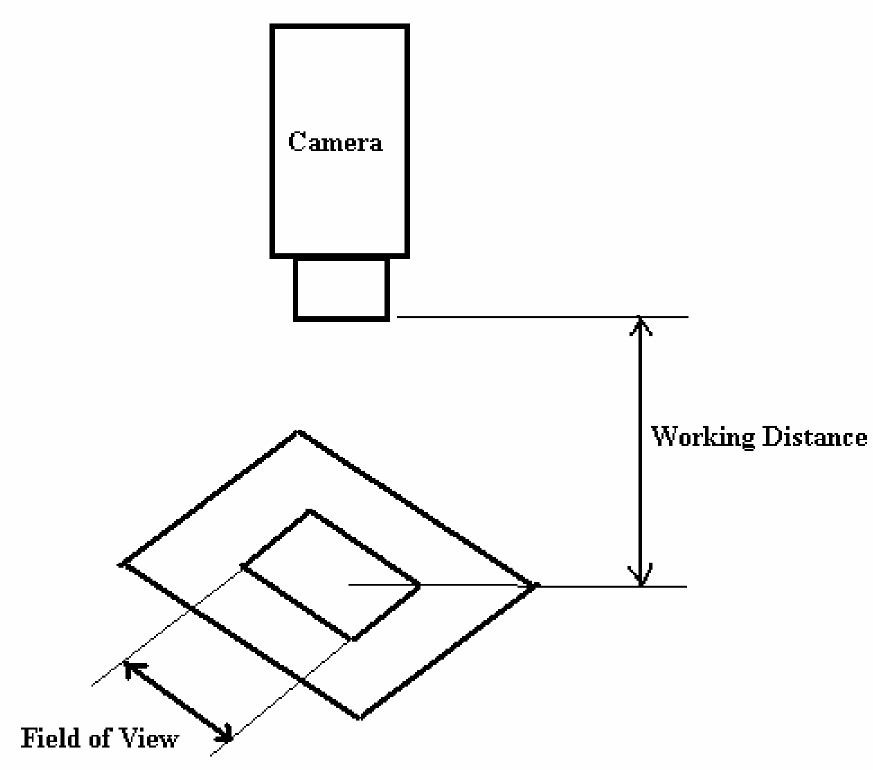


Fig. 3.6. Field of View and Working Distance

Given the sensor size (S_c) and the focal length of the camera (f_c) it is possible to calculate the required WD for a camera given the FOV using Eq. 3.1.

$$\frac{f_c \times FOV}{S_c} = WD \tag{3.1}$$

Therefore, using the specifications of the camera (f_c = 5 cm / S_c = 1/3”), the following table tabulating the theoretical FOV vs. WD values was generated (Table 3.1). In order to be on the safe side, the FOV vs. WD values were also calculated experimentally and these values have also been tabulated in Table 3.2. The experimental values were found to have an exponential relation and the relationship curve has been displayed in Fig. 3.7.

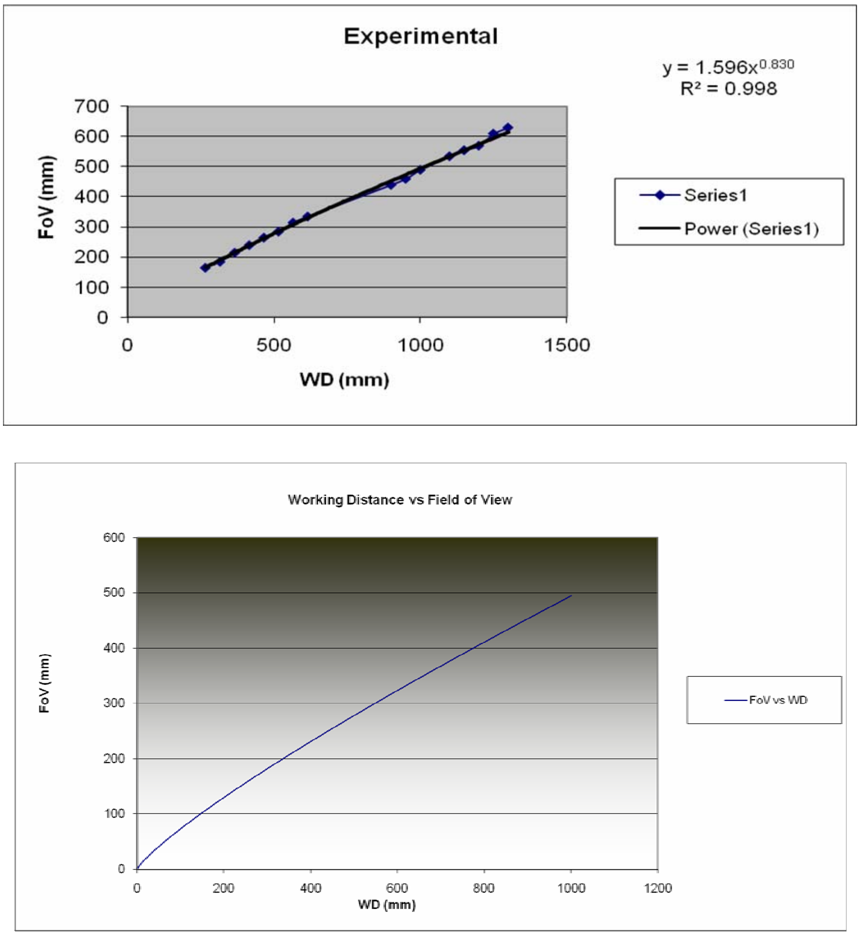


Fig. 3.7. FOV vs WD (Experimental)

No	**WD (mm)**	**FOV(mm)**
1	265	156
2	315	186
3	365	216
4	415	245
5	465	275
6	515	304
7	565	334
8	615	363
9	900	531
10	950	561
11	1000	591
12	1100	650
13	1150	679
14	1200	709
15	1250	738
16	1300	768

Table 3.1. FOV vs WD (Theoretical)

No.	WD (mm)	FoV (mm)
1	265	165
2	315	185
3	365	215
4	415	240
5	465	265
6	515	285
7	565	315
8	615	335
9	900	440
10	950	460
11	1000	490
12	1100	535
13	1150	555
14	1200	570
15	1250	610
16	1300	630

Table 3.2. FOV vs WD (Experimental)

As is clearly delineated in Tables 3.1 and 3.2, the discrepancy between the theoretical values and the experimental values is directly proportional to the WD. Once this was established, the Experimental Curve (Fig. 3.7) was utilized for all further purposes.

3.2 Image Acquisition Trigger

In order to avoid unnecessary delays and computational load on the processor, the entire frame from the camera feed of the trays is not processed continuously. The entire FOV is only processed once triggered by an external or internal stimulus. The selection of the trigger depends on the scenario as is explained in the following sections.

3.2.1 Production Line 1 (PL1)

PL1 utilizes internal stimuli to trigger the acquisition and image processing sequence. It constantly scans one section of the FOV and once that portion satisfies the pre-programmed criteria (which can be easily modified in the GUI interface), it triggers the image processing sequence.

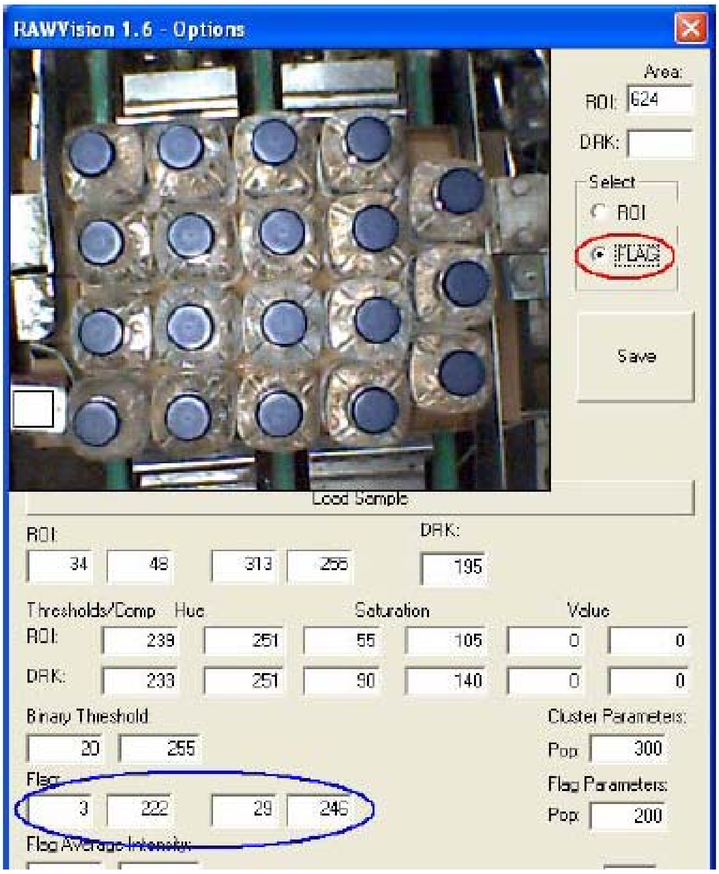


Fig. 3.8. Setting the Trigger (Flag under PL1) parameters

Figure 3.8 displays the Options window of the application written for PL1, namely RAWVision 1.6. The parameters circled in blue represent the coordinates of the rectangle (showed in white in the bottom-left corner of the sample image) which is the region continuously processed to determine whether the tray is stationary and in position.

This was achieved by affixing a white reflector plate on a compacting arm of the packaging machine. The compacting arm closes a flap on the tray sealing it with the aid of glue sprayed previously with a glue gun.

The Options window allows the user to define the minimum area, intensity and size of the FLAG ROI. This makes the trigger definition extremely easy and user-friendly. It is recommended to make the trigger area as small as possible thus reducing computational overhead and processing time.

3.2.2 Production Line 2 (PL2)

PL2 calls for a different approach since the AVI must deal with a stationary camera/moving tray scenario. Delays in image acquisition are not tolerated due to the small FOV (to reduce computational overhead and processing time). In PL2, the image acquisition trigger responds on external stimuli by interfacing the AVI with a photocell output.

The photocell utilized is a part of the production process and by tapping its signals and using it to acquire the image; the AVI acquires the image and processes it only when it is in position. The photocell output has been interfaced with the AVI through the parallel port. Parallel port testing software (Fig. 3.9) was initially written to test the interface and then the code was incorporated into the AVI.

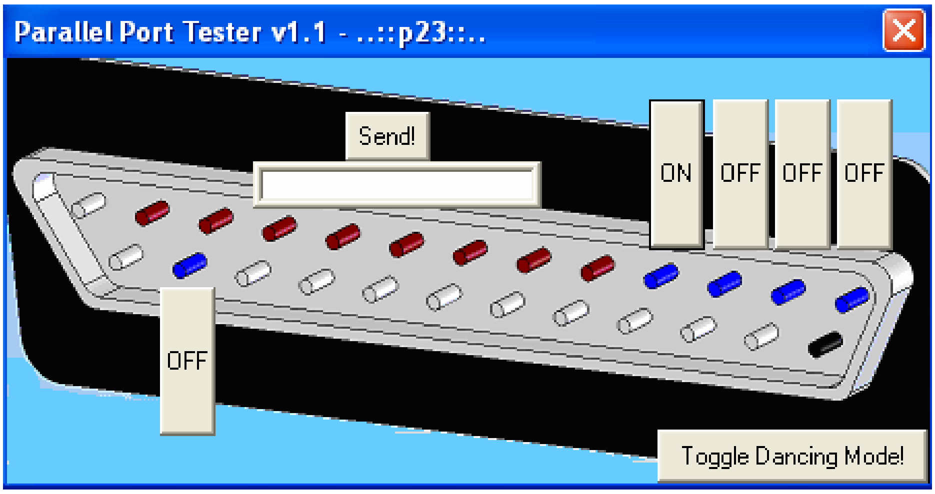


Fig. 3.9. Parallel Port Testing Software

The Parallel Port Testing software was developed in Microsoft Visual C++ 6.0 and was used to test all the I/O pins of the parallel port. The output pins (red) and the input pins (blue) were tested with LEDs and a 5V supply from one of the output pins itself. The photocell from which the signals have been tapped has been displayed in Figure 3.10.

Fig. 3.10 Photocell Assembly

The photocell output was tapped by taking a lead wire from the controller box of the production system. The parallel port interface stores the state of the input pins in a specific memory location (specifically 0x378). Then the AVI reads the data stored in this memory location to ascertain the state of the pins. The standard parallel port circuit diagram and the circuit as well have been displayed below in Figure 3.11.

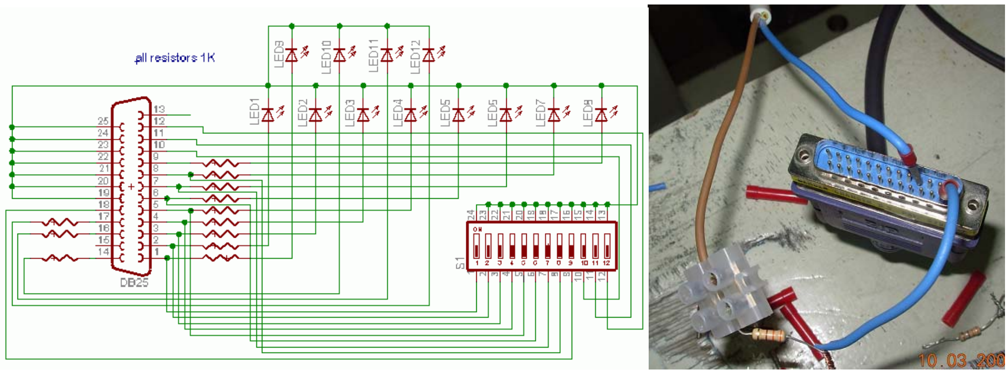


Fig. 3.11. Parallel Port Circuit

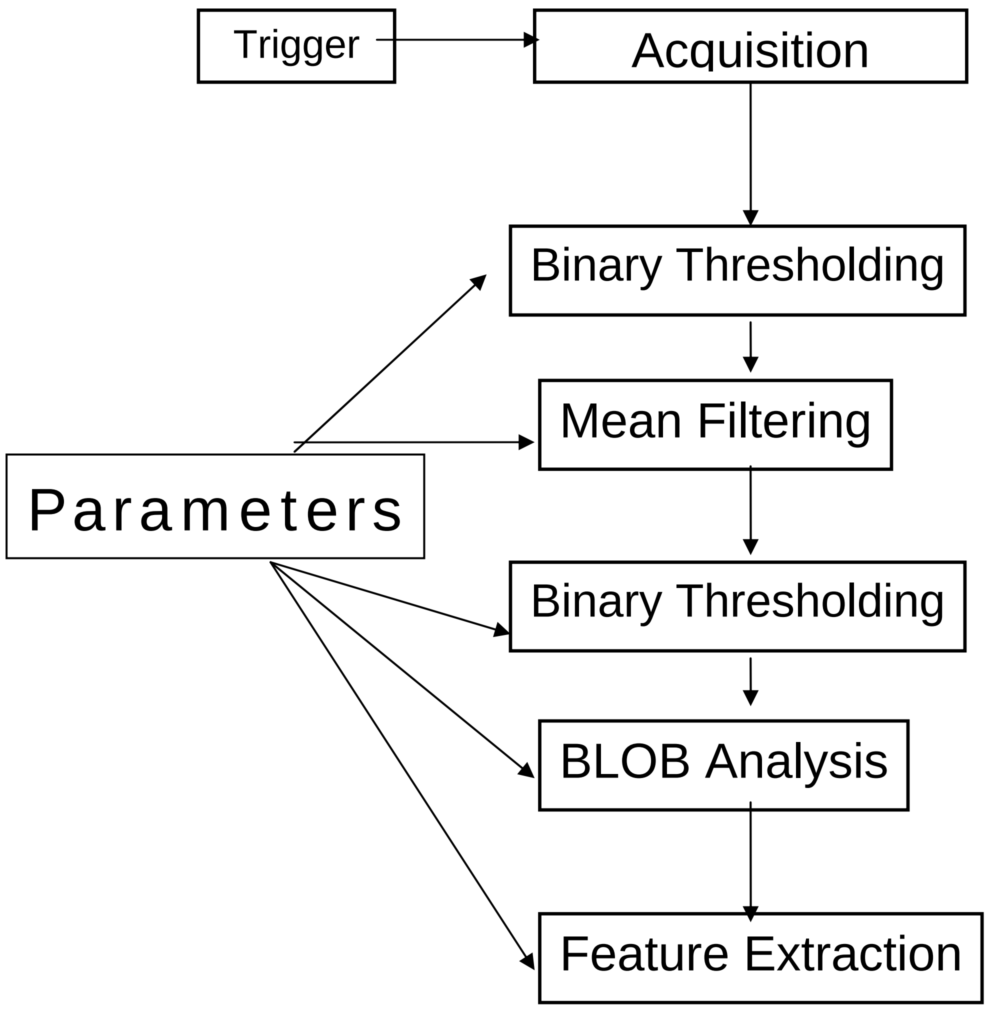
This external stimuli based image acquisition trigger ensures that the image acquisition takes place at precisely the correct time and simultaneously reduces computational overhead and avoids the processing delays encountered in PL1.

CHAPTER 4

METHODOLOGY

This chapter discusses the methodology implemented for the AVI. The various image processing functions designed and utilized in the AVI have been described below accompanied by sample images delineating various stages of the processing portion of the AVI.

The flow chart below depicts the methodology implemented in the image processing portion of this AVI.



4.1 Image Preprocessing

Image preprocessing is vital in the design of any AVI and especially so in this case since it has been implemented in an industrial scenario. The disadvantages experienced due to ineffective lighting will be compensated with the preprocessing algorithms ensuring that the pre-processed image can be easily used for segmentation without any loss in data.

4.1.1 Intensity Histogram Analysis

Intensity Histogram Analysis has been repeatedly used to effectively threshold the acquired image. Using this analysis it is possible to identify key values used in thresholding operations.

In an image processing context, the histogram of an image normally refers to a histogram of the pixel intensity values. It is a graph showing the frequency of occurrence of pixels in an image at each different intensity value found in that image. For an 8-bit grayscale image there are 256 different possible intensities and therefore an intensity histogram will show the distribution of pixels amongst those grayscale values. Histograms can also be taken of colour images; either individual histograms of red, green and blue channels can be taken (or in the case of an HSV model, the hue, saturation and value channels), or a 3-D histogram can be produced, with the three axes representing the red, blue and green channels, and brightness at each point representing the pixel count. The exact output from the operation depends upon the implementation; it maybe a GUI interface graph displaying information about each intensity or a snapshot of the file stored as a device-independent bitmap (DIB) or even a comma-separated value (CSV) file containing the data.

The operation is very simple. The image is scanned in a single pass and a running count of the number of pixels found at each intensity value is kept. This is then used to construct a suitable histogram. If the image is suitable for thresholding then the histogram will be *bi-modal i.e.* the pixel intensities will be clustered around two well separated values. To obtain a suitable threshold value, the intensity values between the peaks of the histogram are generally subjected to analysis. If the distribution is not like this then it is unlikely that a good segmentation can be produced by thresholding.

Due to the sharp contrast between the colors of the bottle caps (blue) and the rest of the image (Fig. 4.1) , the acquired images have proved to be highly bi-modal allowing for effective segmentation through thresholding.

Fig. 4.1 Acquired Image

Histogram analysis software was written using Microsoft Visual C++ 6.0 and was used on the acquired images to ascertain the adequate threshold ranges. Ideal histograms for bi-modal images are displayed below in Fig. 4.2. The first histogram elucidates a single threshold and the second histogram shows multiple thresholding.

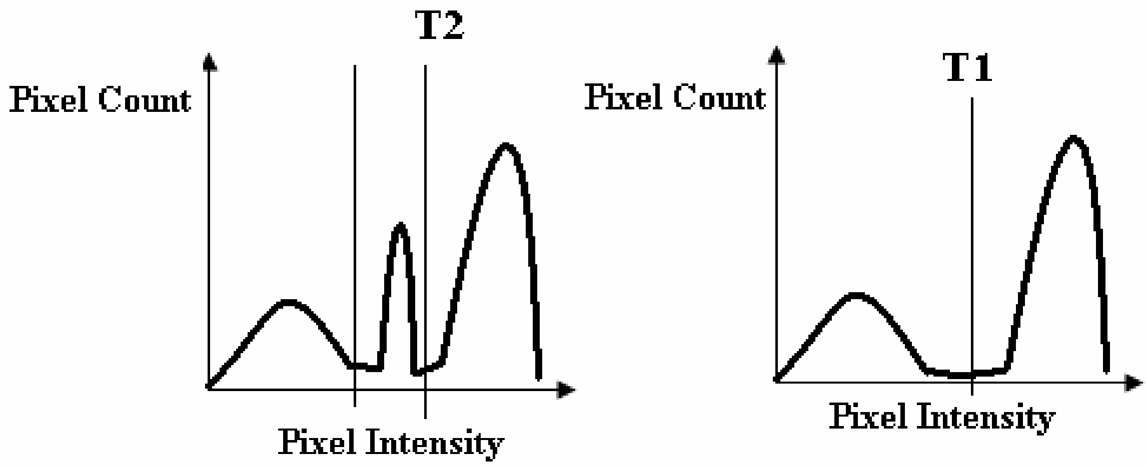


Fig. 4.2 Single and Multiple Thresholding

As observed in Fig. 4.2. the peaks in the histogram indicate optimum threshold ranges. Histogram analysis of several images were conducted and using this data, the optimum ranges for thresholding was obtained.

4.1.2 RGB to HSV Conversion

Several thresholding operations were conducted on the acquired images using the RGB color model but it proved to be highly susceptible to slight changes in lighting and a plethora of other factors. Therefore, an alternative color model, namely the HSV or Hue Saturation Value color model was utilized. The necessity for an RGB to HSV conversion is described below.

4.1.2.1 Windows Bitmap Format

A Bitmap (BMP) file is a bitmapped graphics format used commonly as a simple graphics file format in a variety of operating systems such as Microsoft Windows and OS/2.

Despite the huge file size, the simplicity of BMP and its widespread familiarity in MS Windows and elsewhere, as well as the fact that this format is well-documented and free of patents, makes it a very common format that image processing programs from many operating systems can read and write. BMP files are uncompressed making the read/write and therefore processing operations simpler and faster without any loss to image quality.

The acquired images are in BMP format as the frame grabbed from the video stream is stored as a BMP. Using standard bitmap manupilation classes such as *CBitmap,* it is possible to easily extract the RGB values of every pixel in the bitmap.

The CBitmap class members used in the design of this AVI and a brief description of each have been listed below in Table 4.1.

Class Member	Description
CBitmap	Constructs a CBitmap object.
LoadBitmap	Initializes the object by loading a named bitmap resource from the application's executable file and attaching the bitmap to the object.

GetBitmap	Fills a BITMAP structure with information about the bitmap.
operator HBITMAP	Returns the Windows handle attached to the CBitmap object.
SetBitmapBits	Sets the bits of a bitmap to the specified bit values.
GetBitmapBits	Copies the bits of the specified bitmap into the specified buffer.

Table 4.1. CBitmap Class Members

Using the GetBitmapBits function, it is possible to extract the individual pixel values of the bitmap into a buffer and extract the required RGB values.

4.1.2.2 Conversion Algorithm

The following algorithm has been used to convert the extracted RGB values of a pixel to its corresponding HSV values.

The meaning of all the symbols used in this algorithms and all the following algorithms can be obtained by consulting the nomenclature section of this thesis included in the beginning.

Algorithm:

$$d(x,y) = \left\{ \frac{\frac{(MAX - x)}{6} + \frac{y}{2}}{y} \right\} \tag{4.1}$$

MAX = max(R’,G’,B’)

MIN = min(R’,G’,B’)

Δ= MAX-MIN

IF Δ = 0 THEN H’=S’=0

ELSE

S’= Δ /MAX

V’=MAX

Dr = d(R’, Δ)

Dg = d(G’, Δ)

Db = d(B’, Δ)

IF R’ = MAX

THEN H’ = Db - Dg

ELSE IF G’ = MAX

THEN H’ = $\frac{1}{3} + Dr - Db$

ELSE IF B’ = MAX

$$\text{THEN H'} = \frac{2}{3} + Dg - Dr$$

IF H’ < 0 THEN H’ = H’+1

ELSE IF H’ > 1 THEN H’ = H’- 1

The following images show the different perspectives of the RGB and HSV color model.

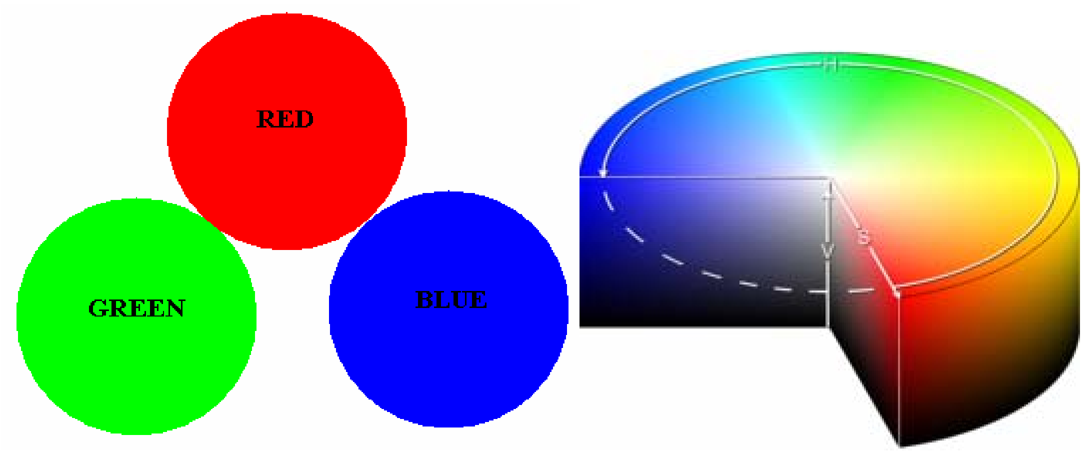


Figure 4.3. RGB and HSV Color Model

The HSV color space could also be visualized as a cylindrical object where the hue varies along the outer circumference of a cylinder, with saturation again varying with distance from the center of a circular cross-section. Value varies from top to bottom. Such a representation might be considered the most mathematically accurate model of the HSV color space; however, in practice the number of visually distinct saturation levels and hues decreases as the value approaches black.

4.1.3 Thresholding

Thresholding is achieved by obtaining the optimum threshold ranges from the histogram analysis. The histogram analysis of a sample acquired image has been displayed below in Figure 4.4

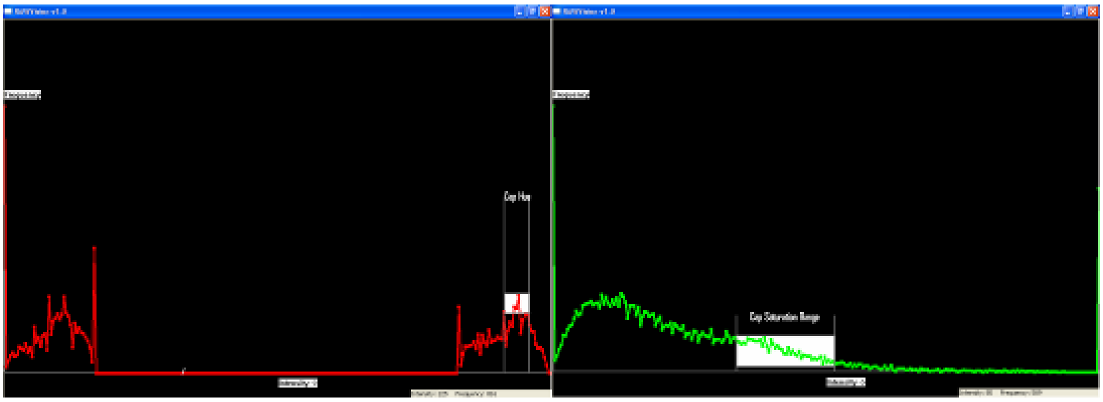


Fig. 4.4 Hue and Saturation Histogram Analysis

In many vision applications, it is useful to be able to separate out the regions of the image corresponding to objects in which we are interested, from the regions of the image that correspond to background. Thresholding often provides an easy and convenient way to perform this segmentation on the basis of the different intensities or colours in the foreground and background regions of an image.

In addition, thresholding allows us to highlight specific areas in the image based on specific criteria such as pixels whose values lie within a specified range, or band of intensities (or colours).

The input to a thresholding operation is typically a greyscale or colour image. In the simplest implementation, the output is a binary image representing the segmentation. Black pixels correspond to background and white pixels correspond to foreground (or vice versa). In this thesis, blue pixels denote the foreground before feature extraction and white pixels denote the extracted bottle cap. In simple implementations, the segmentation is determined by a single parameter known as the intensity threshold. In a single pass, each pixel in the image is compared with this threshold. If the pixel's intensity is higher than the threshold, the pixel is set to, say, white, in the output. If it is less than the threshold, it is set to black. In more sophisticated implementations, multiple thresholds can be specified, so that a band of intensity values can be set to white while everything else is set to black.

Due to non-uniform lighting, part of the image is slightly darkened resulting in different threshold values for that particular region. This hurdle has been overcome by dividing the region of interest (ROI) into two sections and applying different threshold values in both regions.

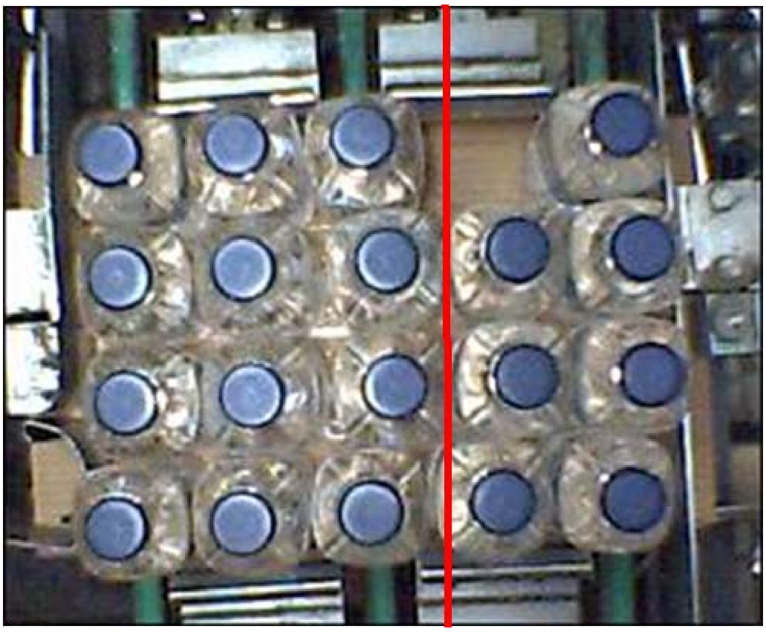


Figure 4.5. Split Region of Interest (ROI)

The following values have been found to be optimal for thresholding the image.

Production Line	Color Component	Normal	Dark
PL1	Hue	$239 \leq H \leq 251$	$55 \leq H \leq 105$
PL1	Saturation	$233 \leq S \leq 251$	$90 \leq S \leq 140$
PL2	Hue	$237 \leq H \leq 245$	$77 \leq H \leq 90$
PL2	Saturation	$230 \leq S \leq 245$	$79 \leq S \leq 90$

Table 4.2. Threshold Values for Preprocessing

Using these values, the acquired image was thresholded to obtain the following image (Fig.4.6.).

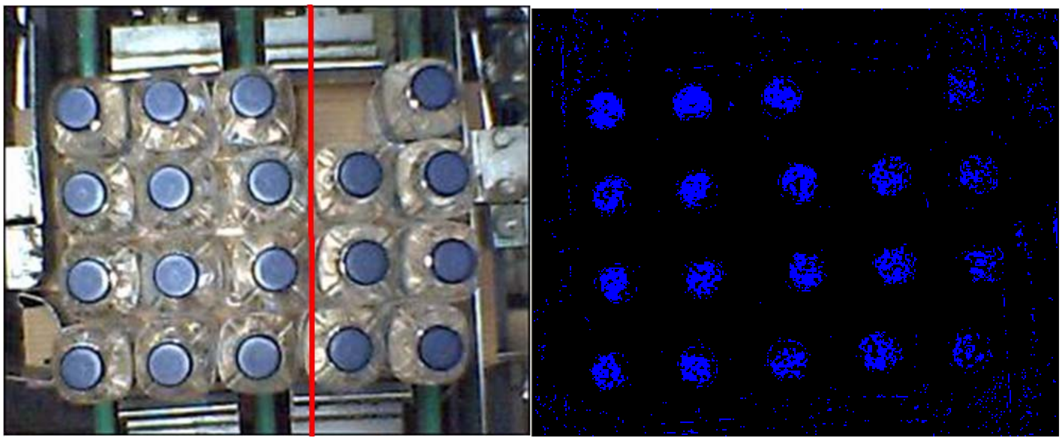


Fig. 4.6. Preprocessing

The ROI alone has not been preprocessed in Fig.4.6 to demonstrate the need for the accurate definition of an ROI. As is apparent from Fig. 4.6, the amount of noise within the ROI is much less than that outside, and hence pre-processing the ROI alone is essential for increasing computational speed as well as reducing computational complexity.

Thresholding was implemented in the acquired image using the following criteria:

$$G_1 = \{T_1 \leq f(m,n):f(m,n) \leq T_2\}$$

$$G_2 = \{ T_1 \geq f(m,n):f(m,n) \geq T_2 \}$$

4.1.4 Convolution

Convolution is a simple mathematical operation which is fundamental to many common image processing operators. Convolution provides a way of `multiplying together' two arrays of numbers, generally of different sizes, but of the same dimensionality, to produce a third array of numbers of the same dimensionality. This can be used in image processing to implement operators whose output pixel values are simple linear combinations of certain input pixel values. In an image processing context, one of the input arrays is normally just a greylevel image.

The second array is usually much smaller, and is also two dimensional (although it may be just a single pixel thick), and is known as the kernel. Figure 4.7 shows an example image and kernel that we will use to illustrate convolution.

I_{11}	I_{12}	I_{13}	I_{14}	I_{15}
I_{21}	I_{22}	I_{23}	I_{24}	I_{25}
I_{31}	I_{32}	I_{33}	I_{34}	I_{35}
I_{41}	I_{42}	I_{43}	I_{44}	I_{45}
I_{51}	I_{52}	I_{53}	I_{54}	I_{55}

$\mathbf{K_{11}}$	$\mathbf{K_{12}}$
$\mathbf{K_{21}}$	$\mathbf{K_{22}}$

Fig. 4.7. Convolution Example: Image and Kernel

The convolution is performed by sliding the kernel over the image, generally starting at the top left corner, so as to move the kernel through all the positions where the kernel fits entirely within the boundaries of the image. (Note that implementations differ in what they do at the edges of images as explained below.) Each kernel position corresponds to a single output pixel, the value of which is calculated by multiplying together the kernel value and the underlying image pixel value for each of the cells in the kernel, and then adding all these numbers together.

Mathematically convolution is defined as:

$$O(i, j) = \sum_{k=1}^{m} \sum_{l=1}^{n} I(i+k-1, j+l-1)\; K(k,l) \tag{4.2}$$

Convolution can be utilized to perform a number of image processing tasks. One such task will be mean filtering, a spatial convolution described in the next section.

4.1.5 Spatial Filtering

Spatial filtering involves the use of a certain element known as *structuring elements.* Structuring elements used in convolution operations are generally called kernels and have a wide variety of applications.

The structuring element consists of a pattern specified as the coordinates of a number of discrete points relative to some origin. Normally Cartesian coordinates are used and so a convenient way of representing the element is as a small image on a rectangular grid. Figure 4.8 shows a number of different structuring elements of various sizes. In each case the origin is marked by a ring around that point. In structuring elements, the origin is not necessarily at the center but just like the fact that most elements are 3x3 matrices, it is commonly followed.

1	1	1
1	**1**	1
1	1	1

	1	
1	**1**	1
	1	

1	1	
1	**0**	
1		0

Fig. 4.8 Structuring Elements

One of the simplest smoothing filters used is the Mean filter. It is easy to implement and computationally light. The Mean filter convolves the following structuring element across the acquired image. In order to achieve optimal smoothing, this is repeated 3 times.

$\frac{1}{9}$	$\frac{1}{9}$	$\frac{1}{9}$
$\frac{1}{9}$	$\frac{1}{9}$	$\frac{1}{9}$
$\frac{1}{9}$	$\frac{1}{9}$	$\frac{1}{9}$

The following image displays the ROI after the smoothing operation has been completed. Note the distinct lack of noise as it has faded into the background leaving only the bottle caps as blurred circular shapes.

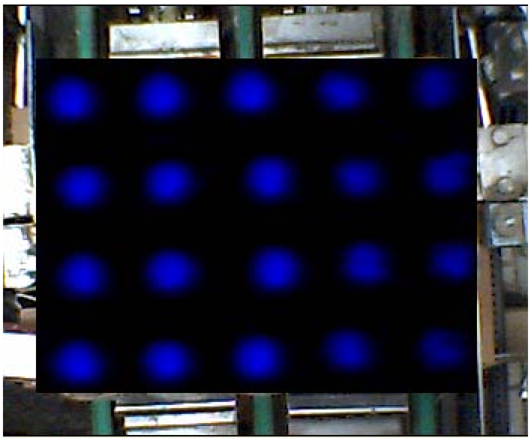


Fig.4.9 Mean Filter

4.2 Segmentation

The smoothed image is subjected to binary thresholding in order to extract the individual bottle caps as Binary Large OBjects (BLOBs). This final thresholding operation ensures that each bottle cap is present as a whole, cohesive binary object in the processed image with minimal noise. As observed from Fig.4.10, the image has been efficiently segmented for feature extraction (BLOB analysis).

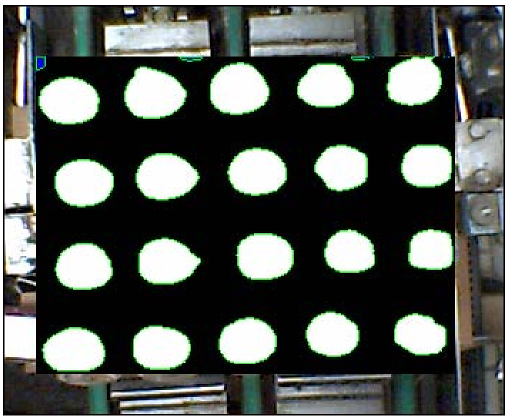


Fig.4.10 Segmentation - Preparation for BLOB Analysis

4.3 Feature Detection

In computer vision and image processing the concept of feature detection refers to methods that aim at computing abstractions of image information and making local decisions at every image point whether there is an image feature of a given type at that point or not. The resulting features will be subsets of the image domain, often in the form of isolated points, continuous curves or connected regions.

4.3.1 Edge Detection

Edges are points where there is a boundary (or an edge) between two image regions. In general, an edge can be of almost arbitrary shape, and may include junctions. In practice, edges are usually defined as sets of points in the image which have a strong gradient magnitude. Furthermore, some common algorithms will then chain high gradient points together to form a more complete description of an edge.

In this AVI, edge detection is performed on a binary image using the following algorithm:

$$\sum_{i=1}^{m} \sum_{j=1}^{n} IF\ I(i,j) \neq I(i+1,j)\ THEN\ I(i,j) = w_c$$
$$\sum_{i=1}^{m} \sum_{j=1}^{n} IF\ I(i,j) \neq I(i,j+1)\ THEN\ I(i,j) = w_c \qquad (4.3)$$

Using this algorithm, the edge points have been detected and stored, but to ascertain the edge about individual blobs and define a minimum enclosing rectangle (MER) for each blob, the following approach is utilized.

4.3.2 BLOB Detection

BLOBs provide a complementary description of image structures in terms of regions, as opposed to corners that are more point-like. Nevertheless, BLOB descriptors often contain a preferred point (a local maximum of an operator response or a center of gravity) which means that many BLOB detectors may also be regarded as interest point operators. BLOB detectors can detect areas in an image which are too smooth to be detected by a corner detector.

Once the edges have been identified, the MER for each BLOB must be defined. This will aid in the feature extraction process, allowing us to count the number of individual bottle caps present in the acquired image and differentiating it from noise and other phenomena.

Firstly, an 8-neighbourhood function is defined to identify the next contour point in the neighbourhood:

$$
\begin{aligned}
N(o_{x,y}) = \; & o_{x-i,y-j} \; if \; o_{x-i,y-j} = w_c \\
& 0 \; if \; o_{x,y} = o_{a,b} \\
& 0 \; if \; D\left(o_{x,y}, N(o_{x,y})\right) > D\min
\end{aligned}
\qquad (4.4)
$$

where (i,j) are integers satisfying the condition specified above and Oa,b is a random contour point selected initially. The function D calculates the Euclidean distance between two points, and Dmin ensures that the contour point under analysis belongs to the contour of the same BLOB.

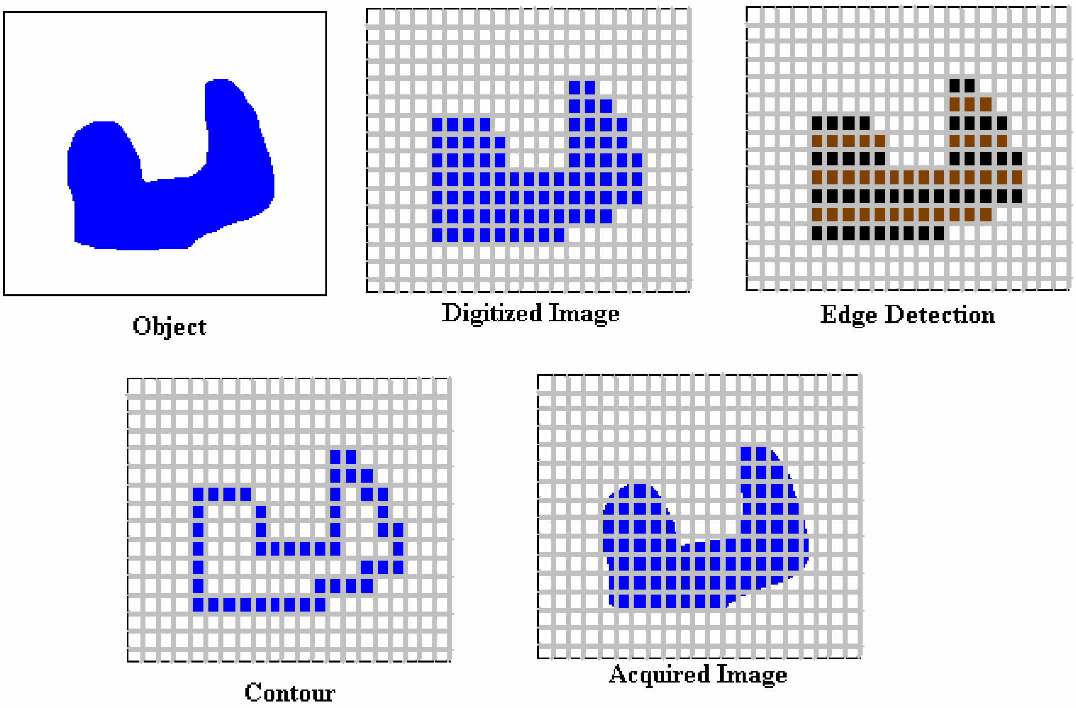


Fig. 4.11 Contour Extraction

Once the edge points for an individual BLOB has been established, its MER is easily defined. Using the MER, feature extraction becomes extremely simple. Fig. 4.12 displays the magnified view of two BLOBs (noise and bottle cap).

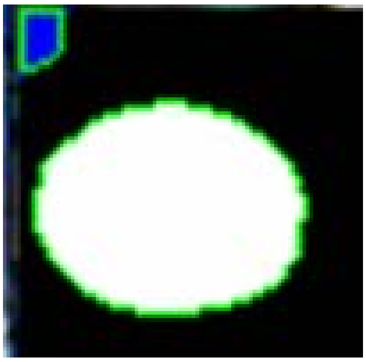


Fig. 4.12 Feature Extraction

Since the MERs for both BLOBs are known, the number of foreground pixels (blue or white) is obtained by checking all the pixels in the MER. If the number of foreground pixels, i.e. area of the BLOB crosses a particular threshold, then the

BLOB is recognized as a bottle cap (white BLOB). Else, it will be treated as noise (blue BLOB). Using this concise and mathematically precise operational definition of the notion of a BLOB makes this feature detection algorithm efficient and robust for BLOB detection.

Once the BLOBs have been identified as a cap or noise, the total number of caps are counted and if it is under a certain threshold (20 bottles in the cases of PL1 and PL2), an alarm goes off alerting the operator to take corrective action.

CHAPTER 5

RESULTS AND DISCUSSION

The aforementioned system was successfully installed and has been under observation on the real-time production line for 3 months and the data have been collected over this time period. The following results were obtained after the AVI was installed on both production lines:

1. A robust, high-speed quality inspection system capable of detecting defects in time to implement corrective measures.
2. A significant decrease in the number of defective trays reaching the market.

Customer feedback and reports from the Quality Control team, it is apparent that the inspection system is working perfectly as not a single complaint has been registered after the installation of this system.

The following table displays the defective trays returned to the company:

Date	Time	Defect
15/01/07	8:41	1 cap
17/01/07	14:44	1 bottle
21/01/07	12:47	1 cap
24/01/07	12:33	2 bottles

Table 5.1. Defective Products

The following table lists the production data gathered from the data logger over a period of 3 months while the inspection system has been in operation.

Date	Approved	Rejected
22-01-2007	4228	104
23-01-2007	4543	80
24-01-2007	4558	111
25-01-2007	4505	115
26-01-2007	4499	44
27-01-2007	4383	51
28-01-2007	4533	107
29-01-2007	4592	56
30-01-2007	4538	57
31-01-2007	4221	46
03-02-2007	3320	33
04-02-2007	4536	57
05-02-2007	4508	23
06-02-2007	4375	30
07-02-2007	3983	76
07-02-2007	3983	76
10-02-2007	4037	34
11-02-2007	6562	41
12-02-2007	6526	30
13-02-2007	6312	25
14-02-2007	5512	32
17-02-2007	4137	40
19-02-2007	4284	115
20-02-2007	7450	74
21-02-2007	7732	41
23-02-2007	4700	38
24-02-2007	5755	21
27-02-2007	6014	23
01-03-2007	6125	36
01-03-2007	5755	21
03-03-2007	5501	17
04-03-2007	6637	98
05-03-2007	6956	71
06-03-2007	6391	49
07-03-2007	5712	57
10-03-2007	3405	12
11-03-2007	4999	19

Table 5.2. Production Data

Fig.5.1 Some Defective Products from January

Upon a request from the company authorities, a production data logger and many other extra features were incorporated into the AVI at no extra cost and minimal effort. This would have only been possible in a computer vision-based AVI which allows for maximum flexibility and massive scope for improvement.

The following images display the acquired image and the processed image in the AVI software (RAWVision) designed with Microsoft Visual C++ 6.0. Figure 5.2 displays images from PL2 which employs RAWVision 1.8 and Figure 5.3 displays images from PL1 which employs RAWVision 1.6. Please note that PL1 is a stationary tray/stationary camera scenario and PL2 is a moving tray/stationary camera scenario which can be clearly observed by the degree of blurring observed in the acquired images.

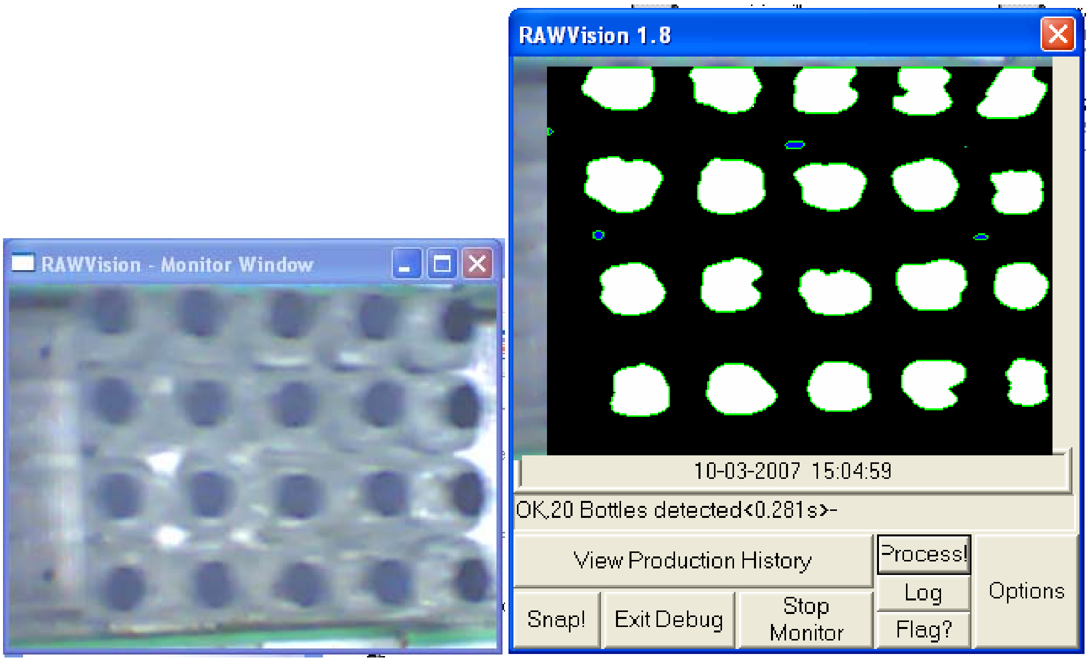


Fig.5.2 PL2

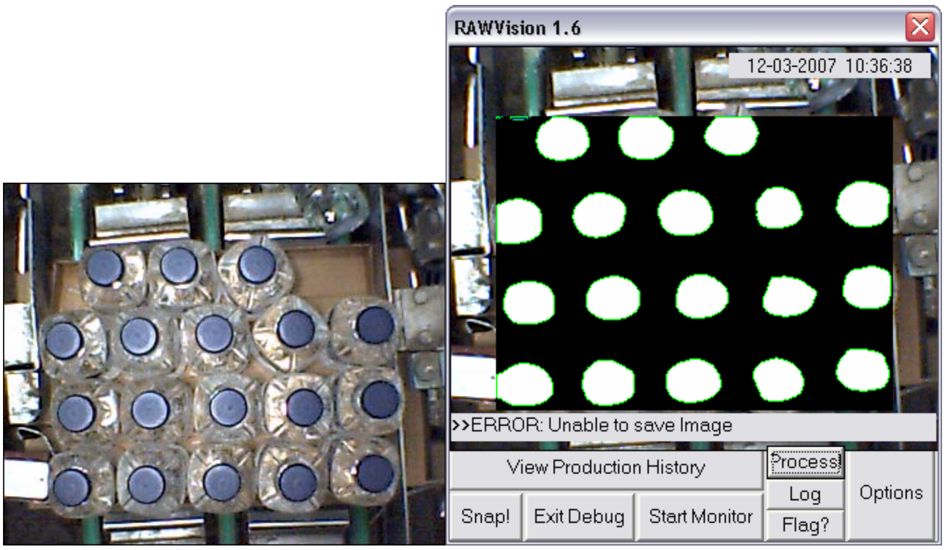


Fig.5.3 PL1

A Production Data Logging system was designed on request and since the AVI is a PC-vision based software, adding new features to it and extended its functionality was extremely simple and at no extra cost at all.

The following image displays the online production information and the production records.

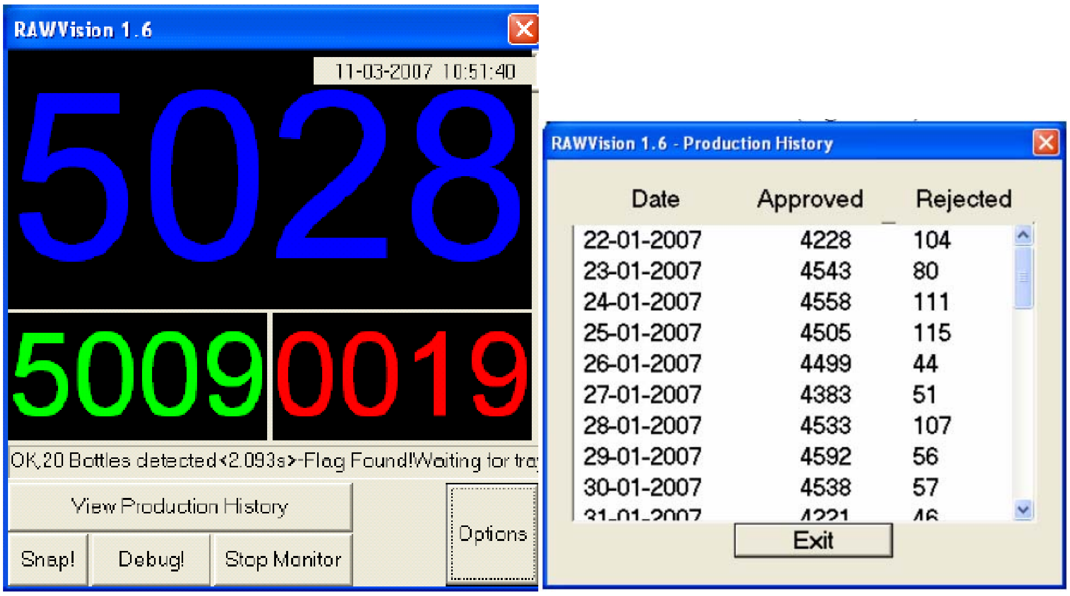


Fig.5.4. Production Data Logging

CHAPTER 6

CONCLUSION

An Automated Visual Inspection (AVI) system for Al Rawdatain Water Bottling Co. (SAK Closed) in the area of bottle packaging has been implemented successfully and satisfactory results have been achieved with the developed system. The execution time for the algorithm is small enough for its use to be feasible in real-time applications as has been demonstrated. User-friendly software with a GUI interface has been developed to implement the algorithms and has also been designed to allow the user to easily change key parameters in several aspects of the image processing sequence.

A production data logger and debugging facility has also been incorporated into the system allowing users to observe production history as well as to modify settings if required. An exhaustive Installation and Maintenance manual has been written (refer Appendix A) and is currently in use.

Quality Analyst reports state that the number of rejections has drastically decreased leading to a decrease in packaging wastage and an increase in product quality and customer goodwill.

Inspection takes around 0.3 seconds per tray on a Pentium IV/3.00 GHz processor and 2.3 seconds on a Pentium I/233 MHz processor. These processing times make it possible for real-time implementation and allow the AVI to run smoothly and detect missing bottles, and missing or askew caps on bottles.

4.1 Scope for Future Work

Future work involves the design and construction of a pick-and-place robot capable of correcting the defective tray without alerting the operators. Using the information gathered from the acquired image, it is possible to obtain the coordinates of the missing location of the bottle and other bottles present in the tray.

Designing a pick-and-place robot with low-cost visual servoing holds great promise and scope for real-time implementation.

REFERENCES

[1] Shingo, S.: ‘Zero quality control: source inspection and the poka-yoke system’ (Productivity Press, 1986)

[2] Groover, M. P.: ‘Automation, production systems and computer-integrated manufacturing’ (Prentice-Hall, 1987)

[3] Juran, J. M.: ‘Quality control handbook (4th edition)’ (McGraw-Hill, 1979)

[4] Kennedy, C. W., Hoffman, E. G., and Bond, S. D.: ‘Inspection and gauging (6th edition)’ (Industrial Press, 1987)

[5] Doif, R. C. (ed.): ‘International encyclopedia of robotics: applications and automation’ (Wiley, 1988)

[6] Boucher, T. O.:‘Computer automation in manufacturing: an introduction’ (Chapman & Hall, 1996)

[7] Panayiotis Panayiotou and Keith Ridgeway: ‘Vision and Quality Improvement’, Manufacturing Engineer, April 2000

[8] K. K. Sreenivasan, M. Srinath, A. Khotanzad: ‘Automated Vision System for Inspection of IC Pads and Bonds’, IEEE Transactions On Components, Hybrids, And Manufacturing Technology, Vol. 16, No. 3, May 1993

[9] A. Beatty, R.G. Gosine, and C.W. de Silva: ‘Recent Developments In The Application Of Computer Vision For Automated Herring Roe Assessment’, IEEE Pix Rim ‘93 Pg. 698 – 701

[10] Francisco J. Barrientos García, Isaac García Incertis, Félix Miguel Trespaderne,Eusebio de la Fuente López, José R. Perán González: 'System for the Production Control and Automatic Packaging of Plastic

Air Sleeve Guides', Proceedings of the Ninth IEEE International Conference on Engineering Complex Computer Systems Navigating Complexity in the e-Engineering Age,1050-4729/04,2004

[11] A. Khotanzad, H. Bannerjee, M. Srinath: 'A Vision System for Inspection of Ball Bonds in Integrated Circuits', 0-8186-2840-5/92, 1992, IEEE

[12] Lu Xiangju, Fan Guoliang, Wang Yunkuan: 'A Robust Barcode Reading Method Based on Image Analysis of a Hierarchical Feature Classification', Proceedings of the 2006 IEEE/RSJ International Conference on Intelligent Robots and Systems October 9 - 15, 2006, Beijing, China

[13] Omron Electronics LLC: Inspection Solutions for the Food & Beverage Industry, Y021-E-02, 10/04/7.5M, Omron Electronics LLC, 2004.

[14] Melles Griot, Lighting Fundamentals, Machine Vision Guide, Melles Griot, 2003.

[15] E. N. Malamas, E. G. M. Petrakis, M. Zervakis, L. Petit, Jean-Didier Legat: ‘A Survey On Industrial Vision Systems, Applications and Tools’, Image and Vision Computing 21 (2003) 171-188

[16] Paul C. Nunnally, Dr. John M. Weiss: 'An Inexpensive Arm for Computer Vision Applications', IEEE Proceedings - 1989 Southeastcon, CH2674-5/89/0000-0001$01.00019891EEE

[17] Giorgio Chivilo, Flavio Mezzaro, Antonio Sgorbissa, Renato Zaccaria: 'The Robotic Attendant: How to Follow a Leader Through Visual Servoing', Proceedings of the 2004 IEEE International Workshop on Robot and Human Interactive Communication at Kurashiki, Okayama Japan September 20-22,2004

[18] W. Zhang, L. M. Koh, Eddie M. C. Wong: ‘Computer Vision System For The Measurement Of Ic Wire-Bond Height’, IEEE TENCON '93 / Beijing

[19] Hamed Sari-Sarraf, and James S. Goddard, Jr.: 'Vision System for On-Loom Fabric Inspection',IEEE Transactions On Industry Applications, Vol. 35, No. 6, November/December 1999

[20] O.T. Strand, M.E. Lowry, S.Y. Lu, D.C. Nelson, D.J. Nikkel, M.D. Pocha, and K.D.Young: 'Automated Fiber Pigtailing Technology',0569-5503/94/0000-l00$03 .00 1994 IEEE

APPENDIX A

RAWVISION INSTALLATION AND MAINTENANCE MANUAL

This manual was generated for the use of the Chief Maintenance Engineer, Mr. John Thomas of the Al Rawdatain Water Bottling Co. (SAK Closed), Kuwait during installation or maintenance of the Automated Visual Inspection System.

RAWVision

Vision-based Quality Inspection System

Installation & Maintenance Manual

Installation & Maintenance:

1. Copy the RAWVision folder into the computer and install the camera drivers from the provided CD and follow the instructions closely.

2. Calculate the optimum height at which the camera should be mounted with the aid of the graph provided below:

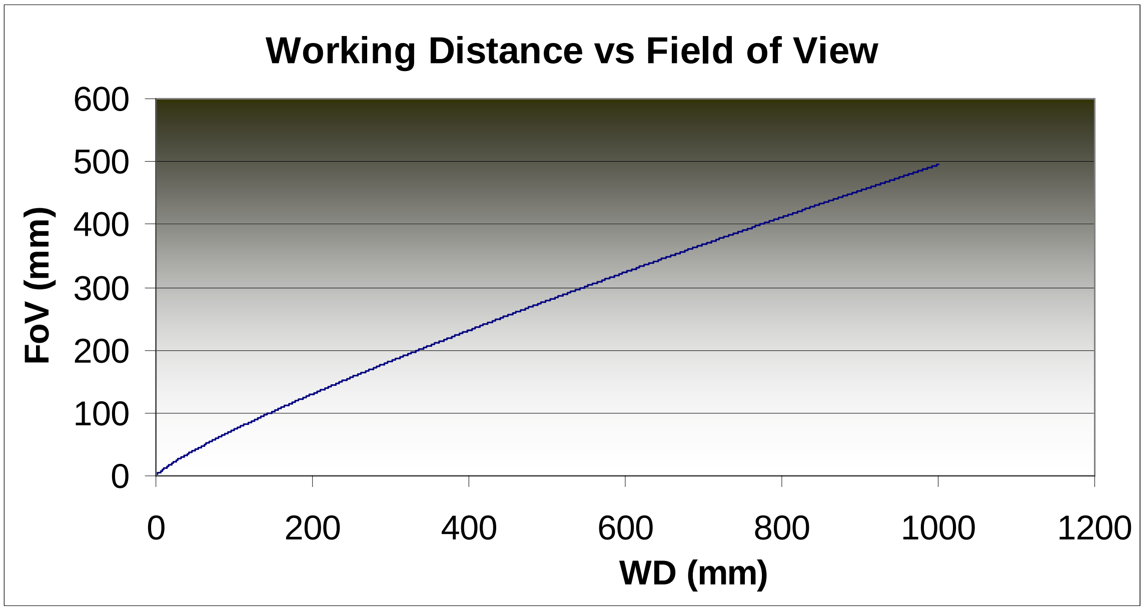


Fig 1.1

The Working Distance is defined as the perpendicular distance of the camera lens above the bottle caps and the Field of View is defined as the maximum width visible in the camera's field of view. Use this graph to obtain an approximate Working Distance value.

3. Once the camera is installed connect it to the computer via the USB extension cable and start the RAWVision software. Adjust the camera so that the image feed looks similar to Figure 1.2.

The adjustments can be made by modifying the position of the camera with the aid of the camera mount as displayed in Figure 1.3.

Fig 1.2

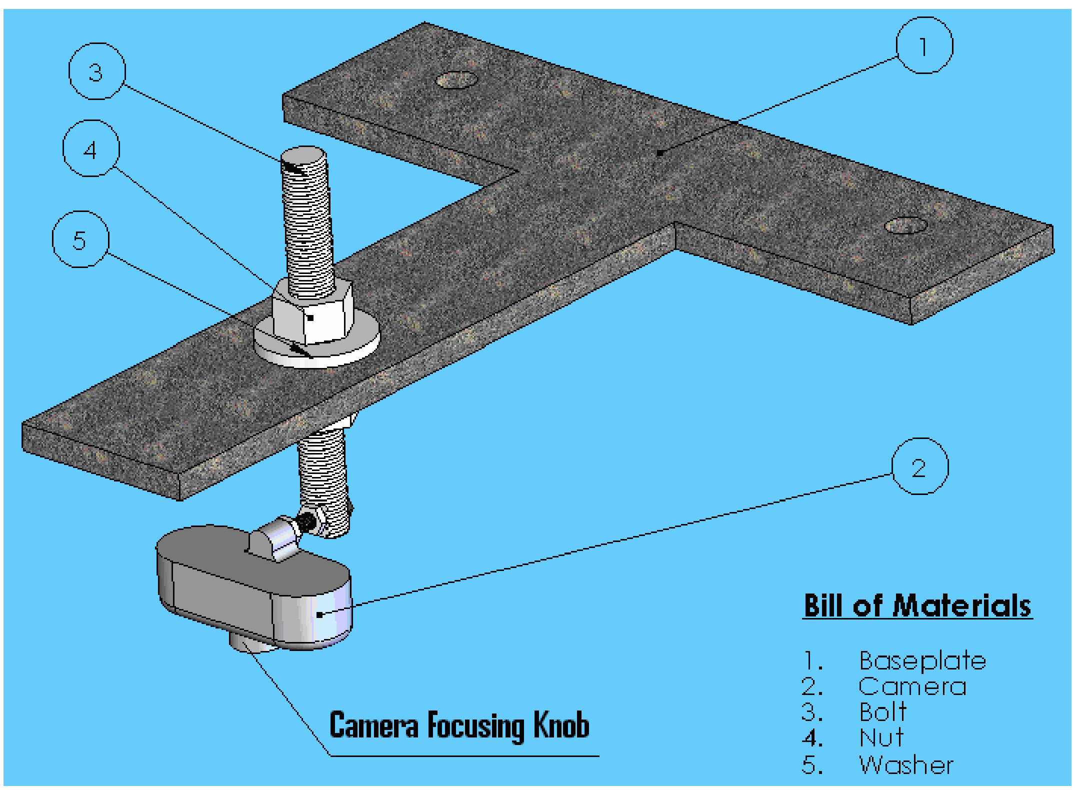


Fig 1.3

4. Once the camera has been positioned correctly, click on the **Snap!** Button on the interface to take a snapshot of the image from the camera feed. You can use any software to do this as long as the image size is set to RGB 352 x 288. This image will be used to calibrate the system and should look like the image in Fig 1.2.

5. Click on the **Options** button to obtain the Options window (Fig 1.4) and click on the **Load Sample** button and select the image obtained from *Step 4.* The Options window should look like Fig 1.5 now.

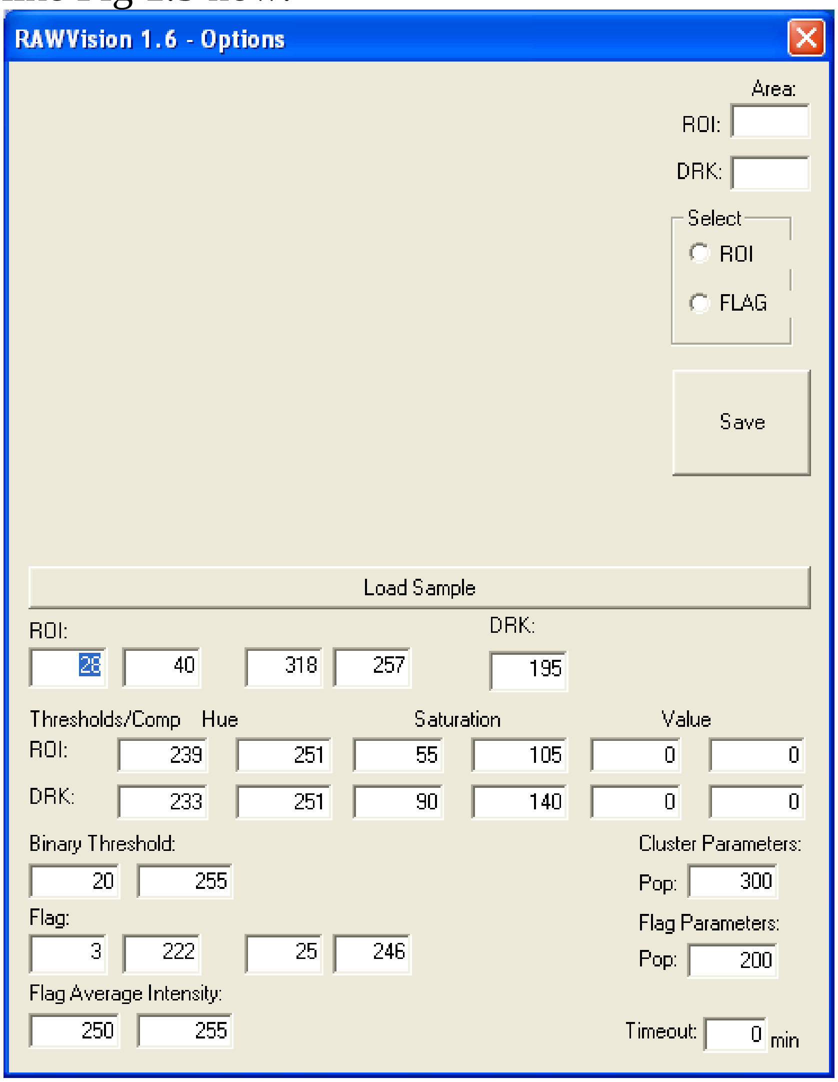


Fig 1.4

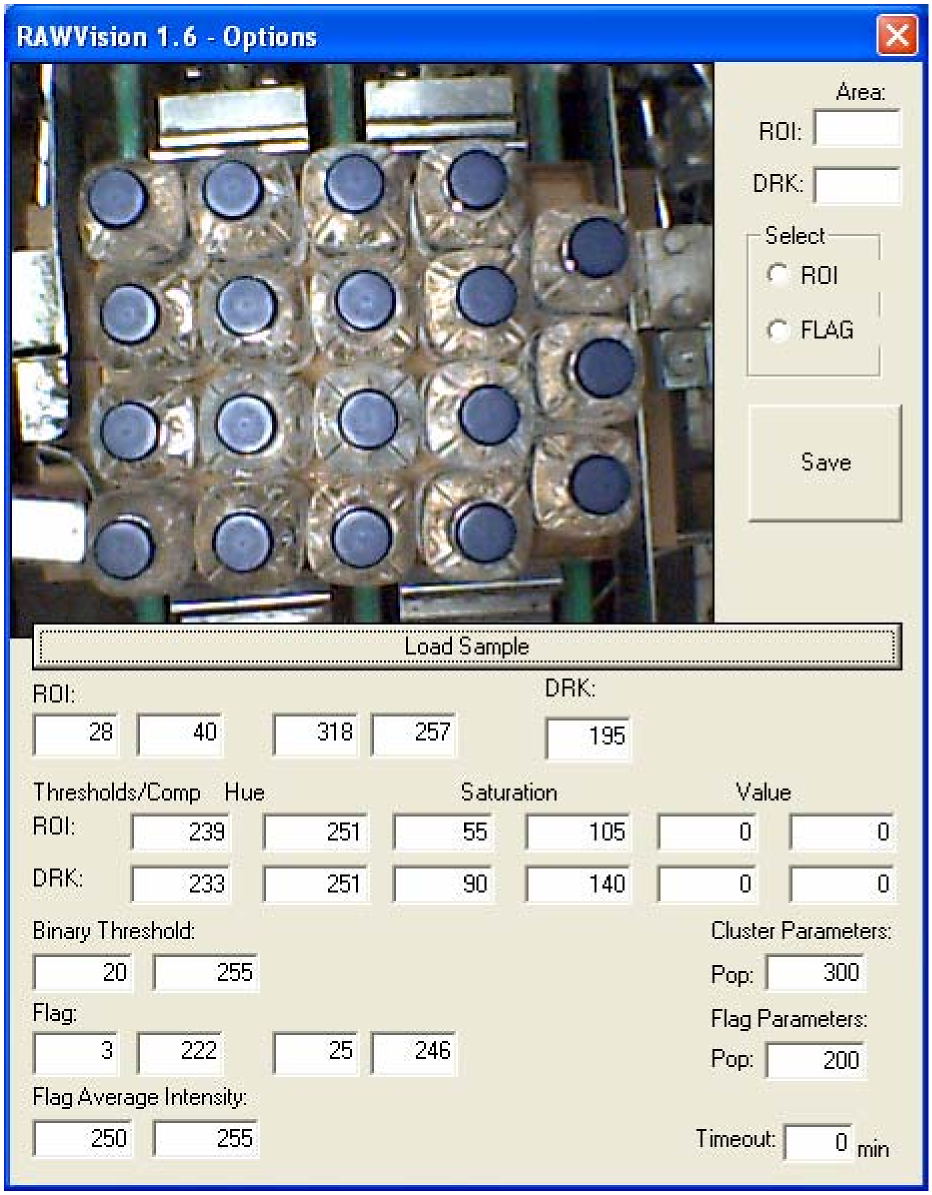


Fig 1.5

6. Check the ROI (Region of Interest) radio button (circled in red in Fig 1.6) and left-click the top-left corner and while keeping it pressed, drag the cursor to the bottom-right corner of the smallest window containing all the bottles and let go of the button. Once you have clicked a white rectangle will form as long as you keep the left button pressed indicating the window you have chosen and the final ROI selected will be displayed. Please note that values in the ROI coordinates (circled in blue) will change and that the first two values represent the (x,y) coordinates the top-left corner and next

two values represent the (x,y) coordinates bottom-right corner of the ROI. The ROI is displayed in Fig 1.6.

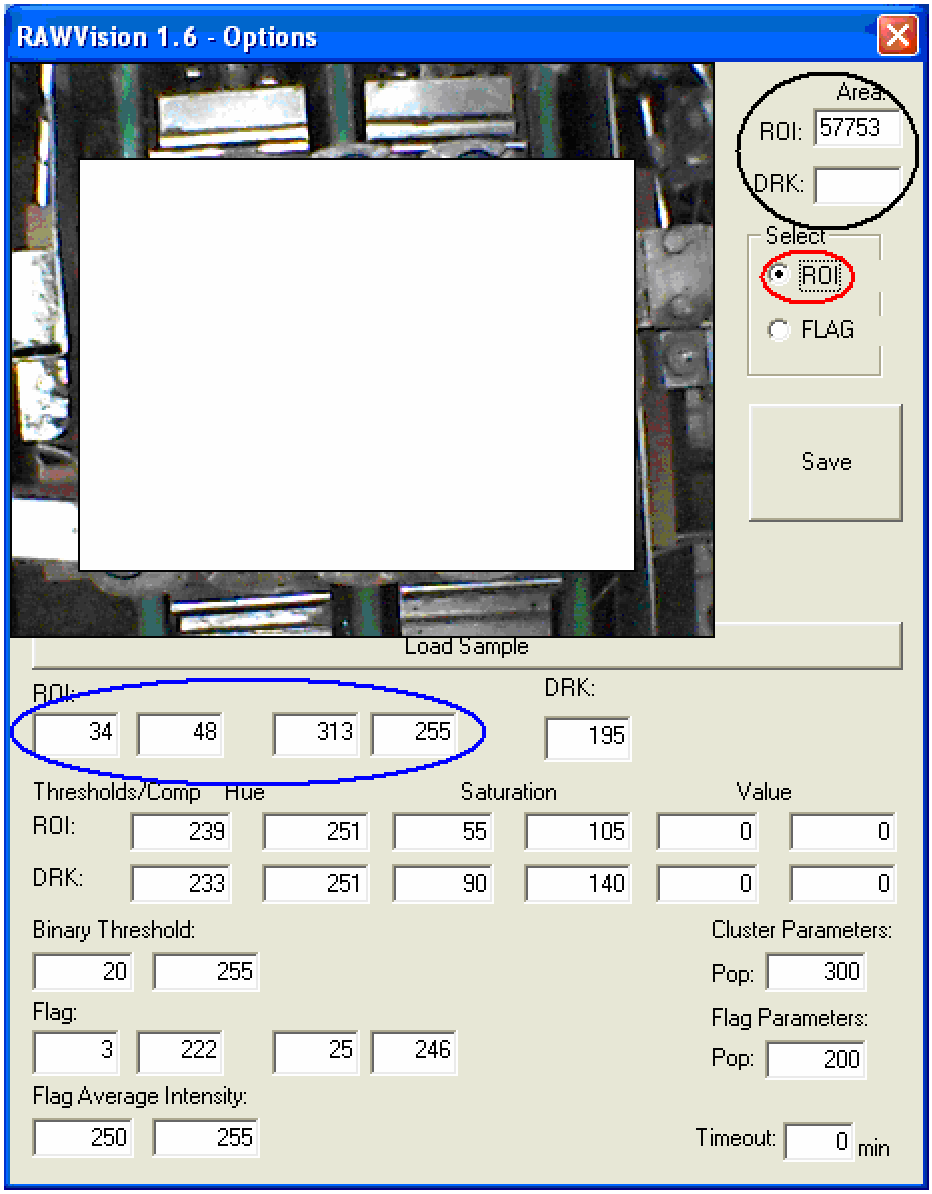


Fig 1.6

7. Similarly, click on the **Flag** button and select an area inside the flag as shown in Fig 1.7.

   Note that the area of the ROI is displayed in the **Area** box (circled in black in Figure 1.6) and should be positive and as small as possible.

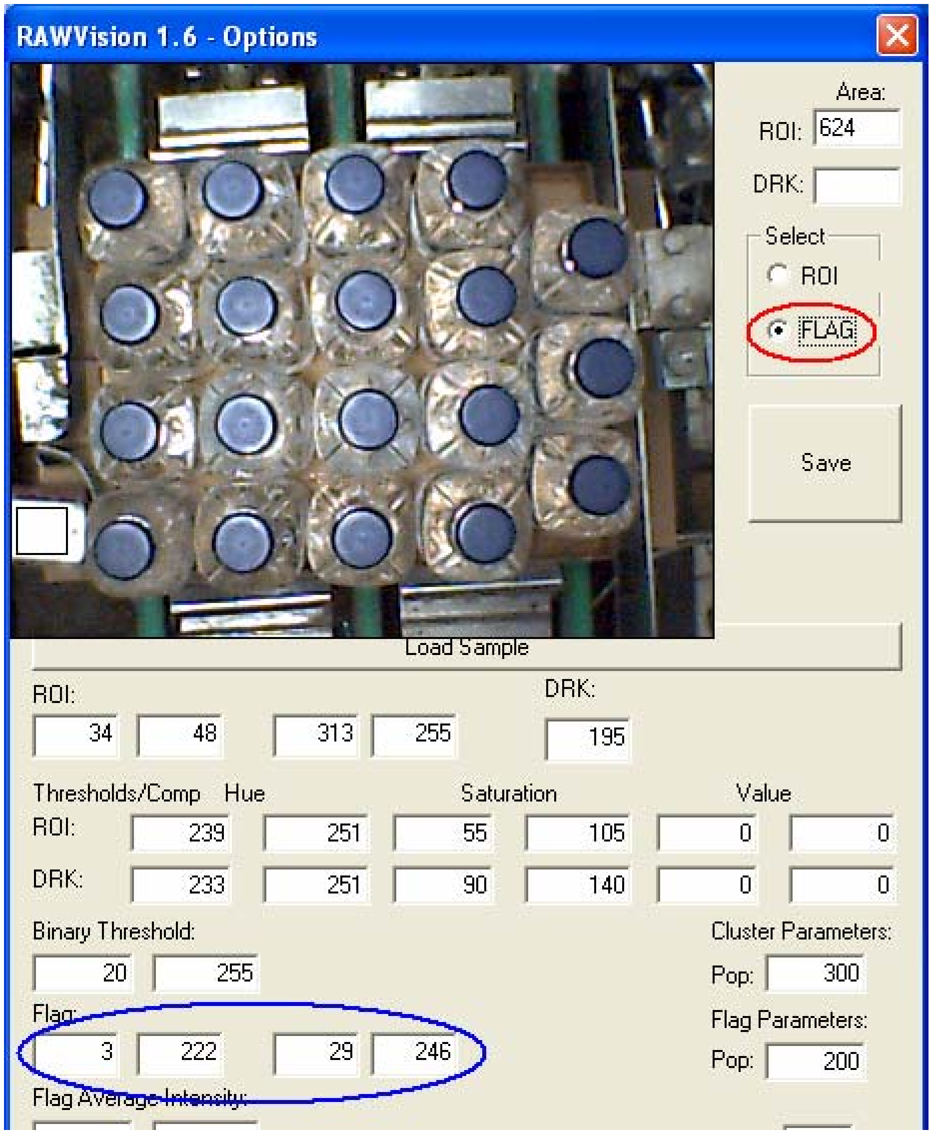


Fig 1.7

8. If you observe the last two columns of the array of bottles, you will notice that they are slightly darker than the rest of the image. Select the darker region by **right-clicking** on any point on the red line (shown in figure 1.8). You will notice that the **DRK** value changes as you right-click at different points.

9. Please note that the red line, red & blue ellipses are displayed in the images above for illustration purposes only and **will not** be present in the Options window.

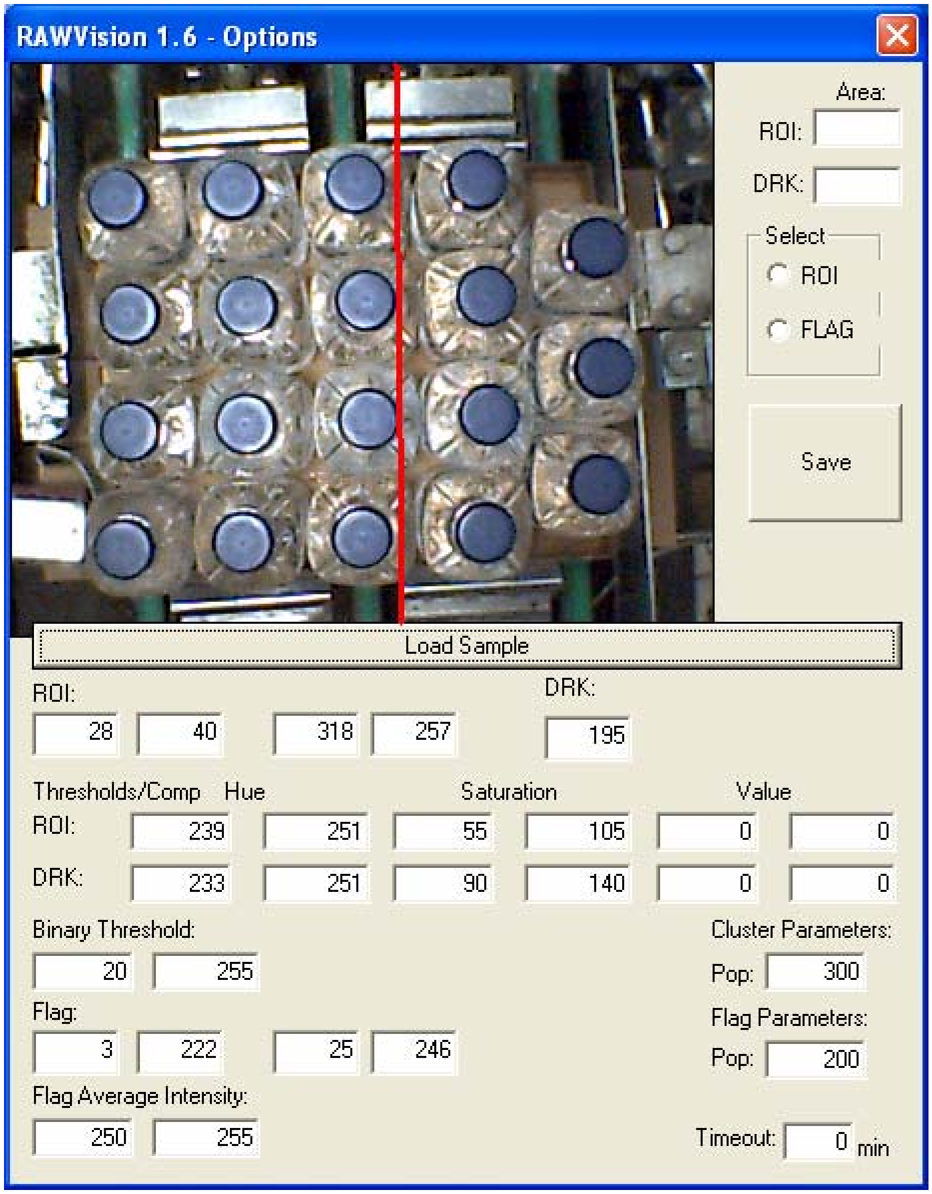


Fig 1.8

10. Now that the ROI, FLAG and DRK regions have been specified, click on the **Debug!** button of the main window to start debugging mode. Debugging mode stores all images where an insufficient number of bottles are present and also allows you to view the processed image on the screen instead of production statistics. It also allows you to perform debugging functions like **Process!, Log,** and **Flag?.** For more information, refer the **Console** section of this manual. The

console view in and out of debugging mode has been displayed in Figure 1.9.

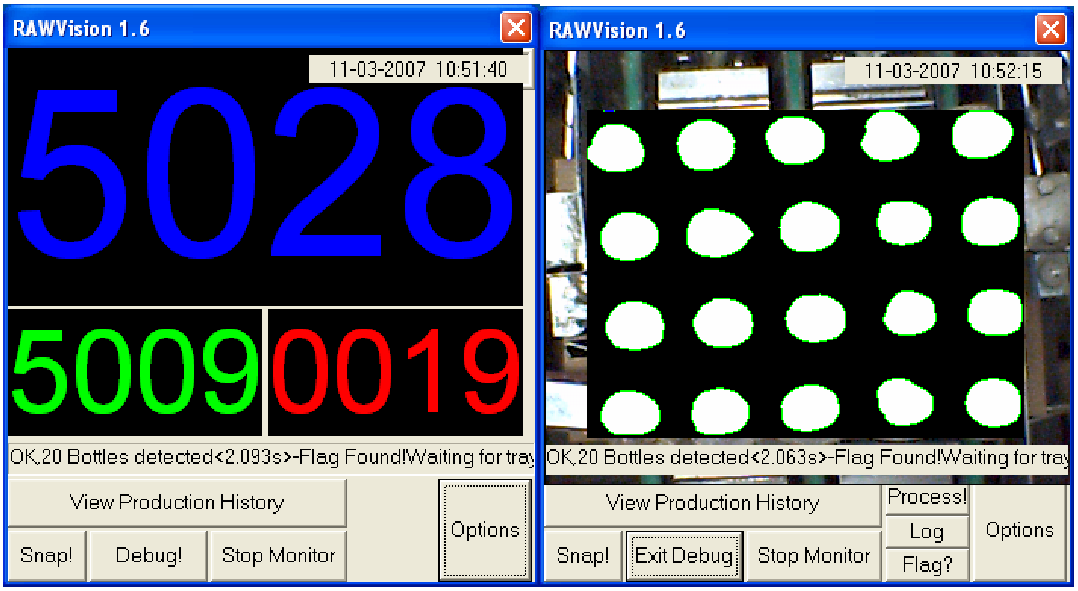


Fig 1.9

Now, upon observing the processed image it is apparent that the ROI & DRK regions have been selected correctly since the white blobs (representing the bottle caps) are distinct and not touching each other. The presence of blue blobs indicates noise.

DO NOT FORGET TO HIT THE *SAVE* BUTTON IN ORDER TO SAVE THE NEW SETTINGS

In the event that the lens has to be cleaned, follow these steps after cleaning the lens:

a. Gently turn the focusing knob on the camera (refer Figure 1.3) until the image on the screen is sharp.
b. Follow Steps 4 – 10.
c. If the white blobs are touching each other or bottle caps are not detected, click on the **Options** button and change the values of the **Hue** and **Saturation** (circled in red in Figure 1.10) settings of the ROI until the first 4 columns of bottles are processed properly. You can process the snapshot you took in Step 4 by clicking the **Process!** button in **Debug** mode.

d. Repeat Step c for the **Hue** and **Saturation** (circled in black in Figure 1.10) settings of the DRK region until the last two columns of bottles are processed properly.
e. If bottle caps are represented as blue blobs and not as white blobs this is because the size of the blob is small. To alter the threshold for the size of the blob (crossing the threshold implies that the blob represents a bottle cap) change the **Cluster Parameters: Pop** value (circled in blue in Figure 1.10). Please modify this value as a last resort; changing the Hue & Saturation values is more than sufficient.
f. Please note that the values should not be changed drastically and only in small increments of 1. If by any chance, the old settings are lost, delete the RAWVision folder and copy it onto the computer again.

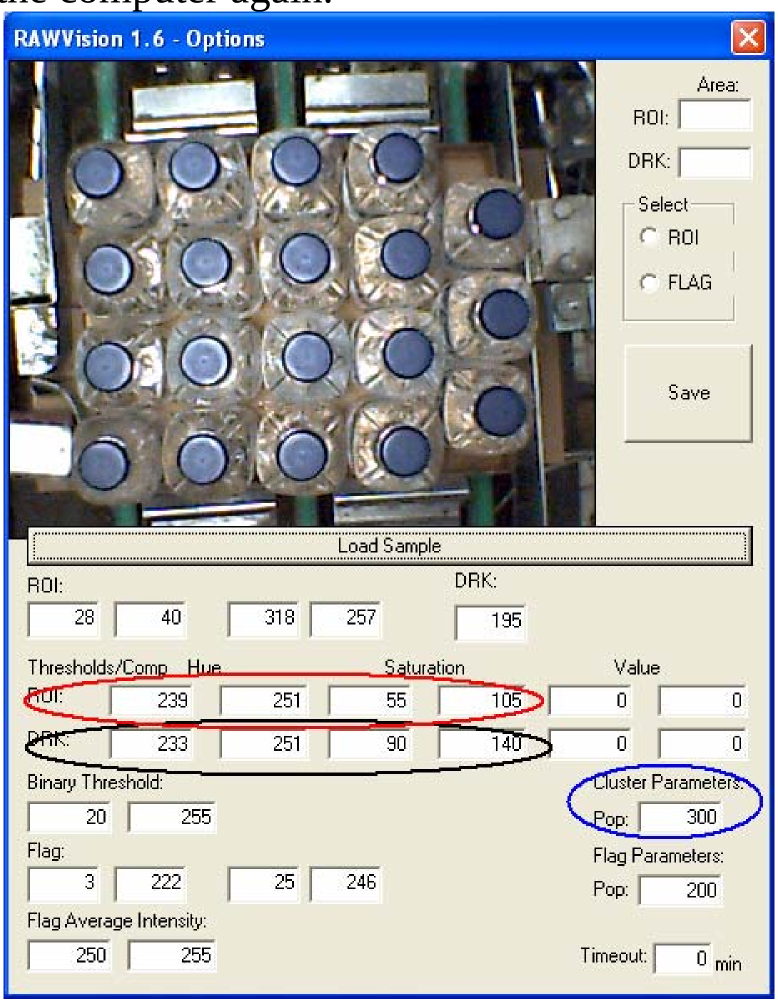


Fig 1.10

Console:

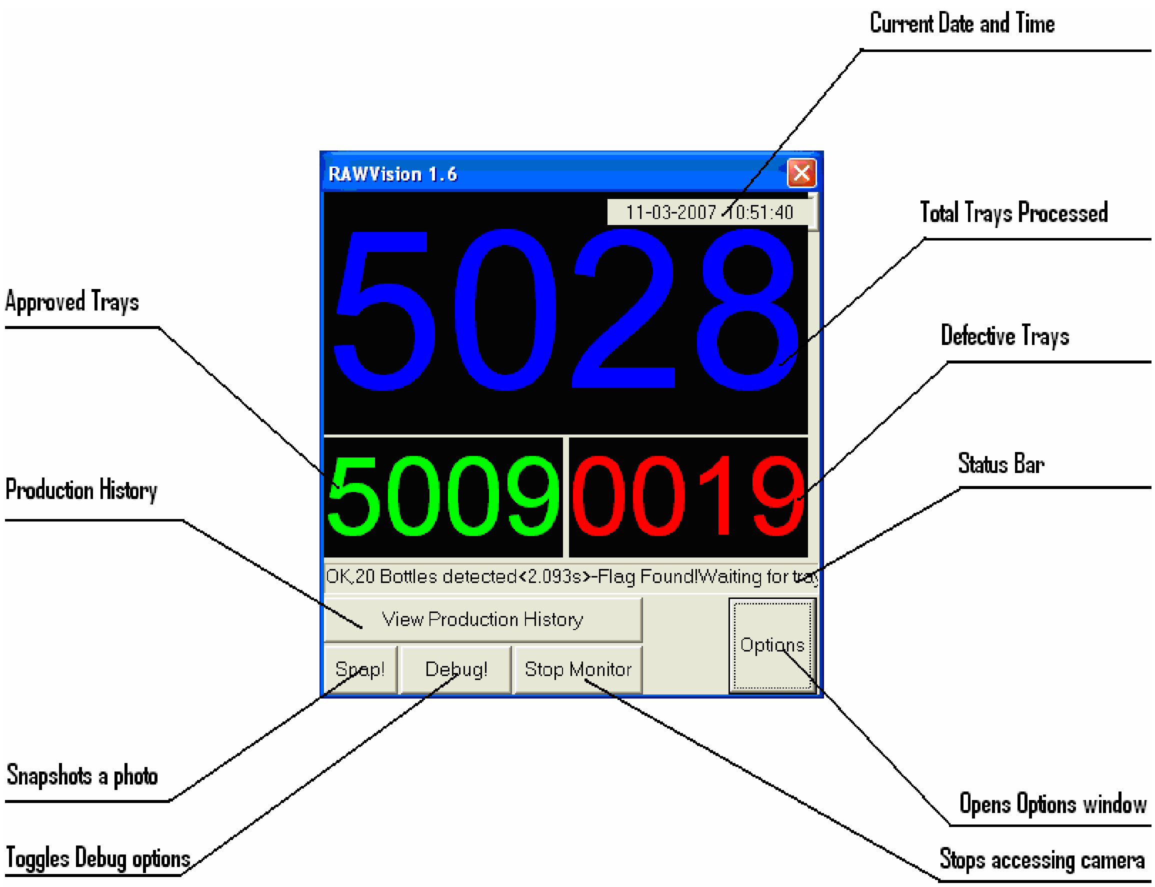


Fig 2.1

- **Status Bar**

The status bar indicates the status of the operations RAWVision is executing at the moment. It contains lots of useful information and can be used for troubleshooting.

- **Snap!**

This button takes a snapshot of the image and requests the user to select the location of the image. Once the location has been established, clicking **Save** will capture the image and save it in the desired location.

- **Stop Monitor**

This closes the camera preview window and stops the program from accessing the camera. This is useful while trying to run the program when camera feed is not required.

- **View Production History**

This button opens the **Production History** window which contains Production statistics (Figure 2.2).

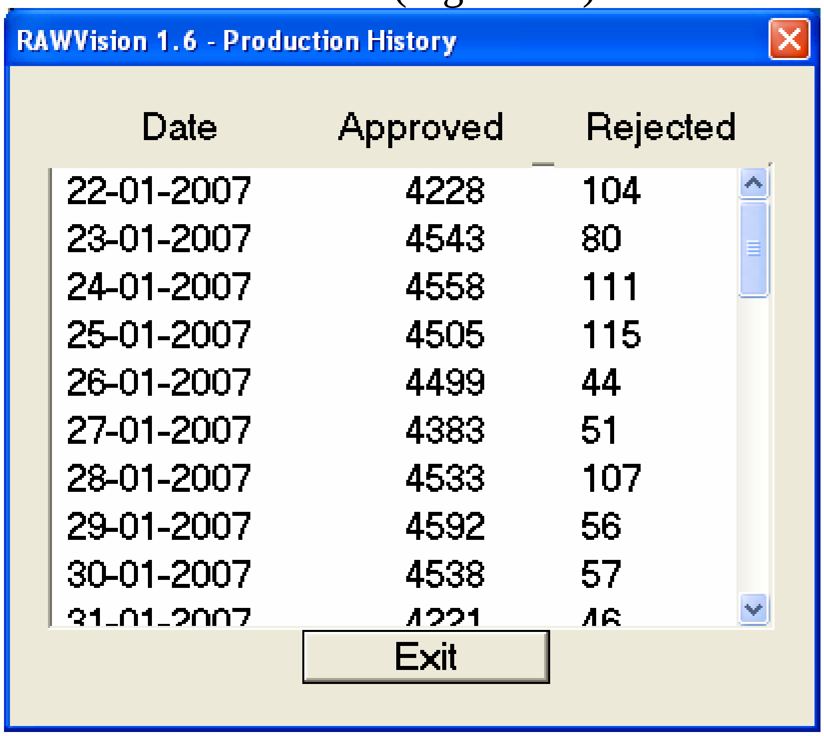


Fig 2.2

- **Debug**

This toggles the Debugging options which are normally kept hidden. It also displays the processed image after every operation instead of production statistics. The three debugging options are:

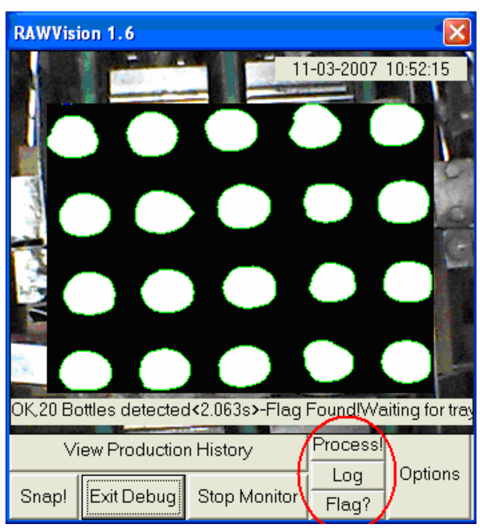


Fig 2.3

- **Process!**

This button allows you to select a saved image and process it with the given parameters. This is useful when calibrating the system. The processed image is displayed over the status bar.

- **Log**

This displays the activity log of the software.

- **Flag?**

This button allows you to select a saved image and check if a flag is present. This is useful when testing the flag settings. The processed image is displayed above, over the status bar with the Flag area in red. Please note that the processed area is set in the Options window (refer to **Installation & Maintenance**).

- **Options**

The options window is by far the most important part of RAWVision and can be used to alter various settings (Figure 2.4).

  - **Area: ROI ; DRK**

  Displays the area of the ROI (Region of Interest: smallest possible rectangle containing all the bottles) and DRK (area of the darkened area or the last two columns of bottles) while setting the ROI and DRK coordinates (refer Steps 5-10 in the **Installation & Maintenance** section).

  - **Select: ROI ; FLAG**

  Allows you to select the smallest possible rectangle to define the ROI and FLAG area (refer Steps 5-10 in the **Installation & Maintenance** section).

  - **Save**

  Saves the settings to a file (*beta.def).*

  - **Load Sample**

  Loads a saved image to set the ROI, FLAG & DRK settings in the Options window. (refer Step 5 in the **Installation & Maintenance** section).

- **ROI: x1,y1 x2,y2**

Displays the coordinates of the top-left corner (x1,y1) and bottom-right corner (x2,y2) of the ROI. In Figure 2.4, the coordinates would be (28,40) and (318, 257).

- **DRK: x1**

Displays the top-coordinate of the darkened region (last 2 columns). With this value, the darkened region is defined with the ROI coordinates. In Fig 2.4, the darkened region would be defined by (195,40) and (318,257). This is made clear in Figure 2.5.

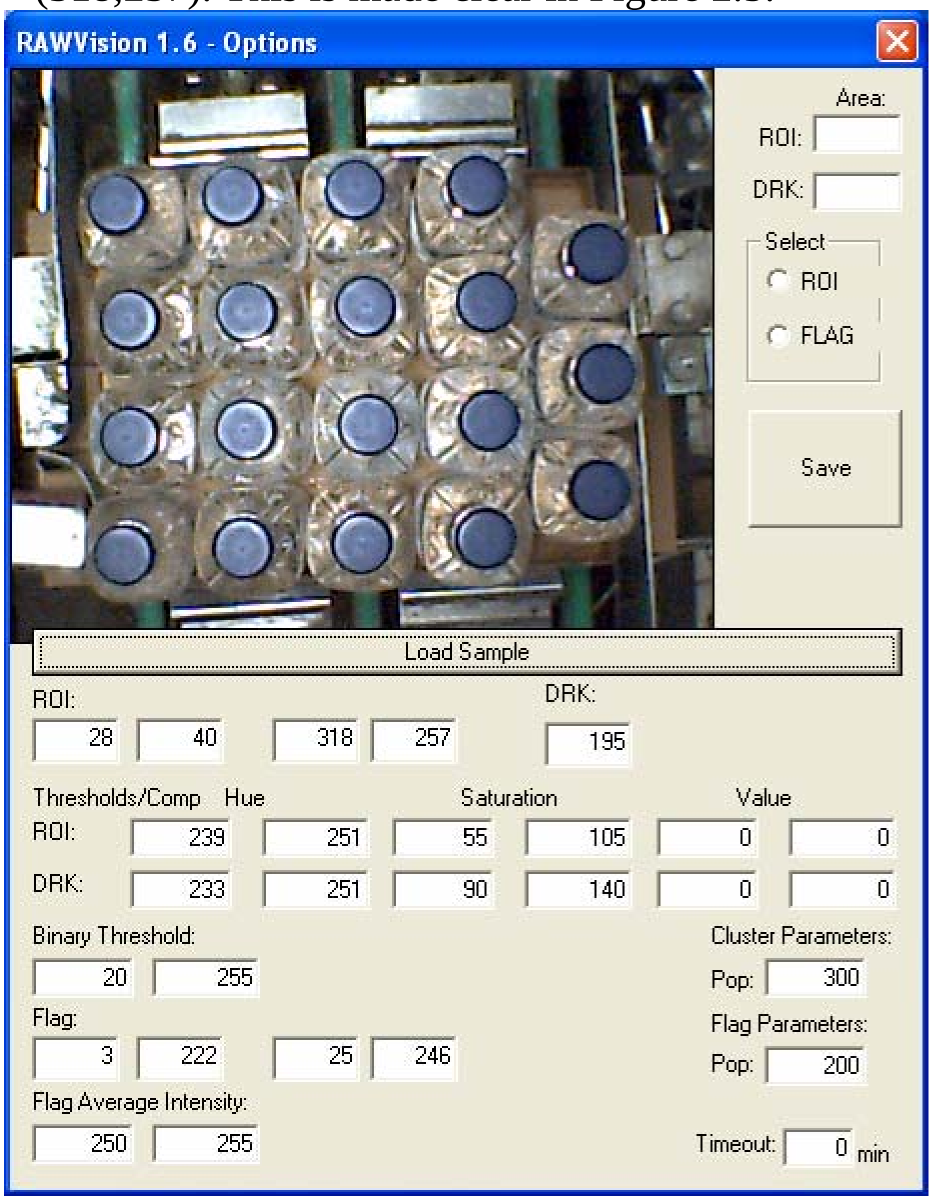


Fig 2.4

Figure 2.5 displays the ROI enclosed in the area enclosed by the red & blue rectangles and the darkened region enclosed in the red rectangle. The coordinates are taken from the settings in Fig 2.4.

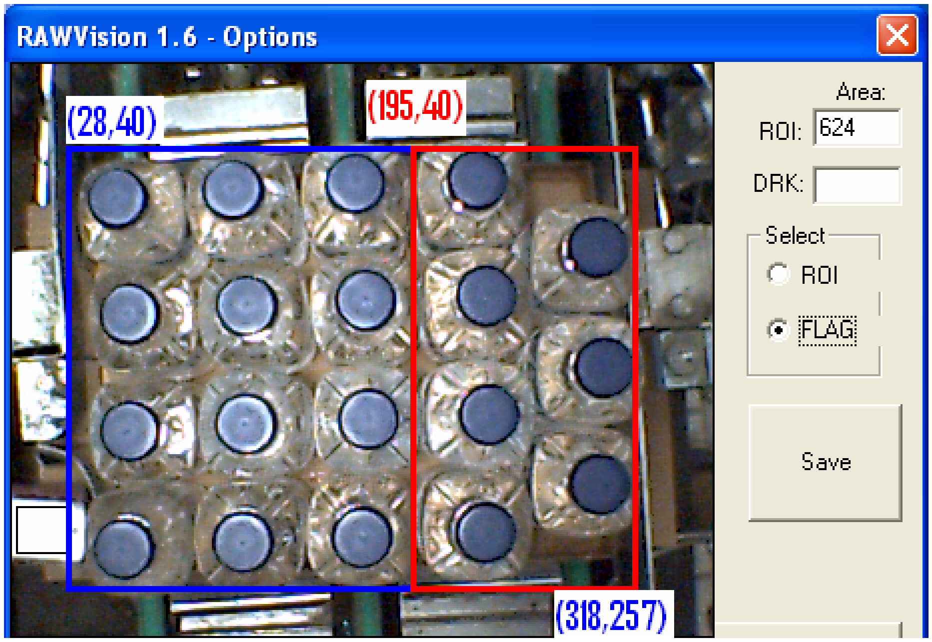


Fig 2.5

- **Thresholds/Comp: Hue Saturation Value**

ROI & DRK: The first two values indicate the range of values indicating the *Hue* of the color of the bottle caps in the ROI and the second set of values represent the range of the *Saturation* values. The second row of values indicate the same for the DRK region. Pay no attention to the 5th & 6th values in each row indicating the *Value* color setting (always set to 0).

- **Binary Threshold**

Represents the range of values indicating the binary threshold for the segmentation of the blurred image. Do not alter these values without consultation. Please note that increasing the first value (indicated by 20 in Fig 2.4) will reduce the size of the blobs.

- **Cluster Parameters: Pop**

Represents the threshold value for the area of a blob. If the area of a blob exceeds this value, it is recognized as a bottle cap (indicated by the color white). If it is less than this value, it is recognized as noise (indicated by the color blue). It is recommended that this value be changed only with consultation.

- **Flag: x1,y1 x2,y2**

Displays the coordinates of the top-left corner (x1,y1) and bottom-right corner (x2,y2) of the FLAG. In Figure 2.4, the coordinates would be (3,222) and (25,246).

- **Flag Average Intensity:**

These values indicate the threshold range for the flag. A value of 0 indicates black, and 255 white. This is why the values have been set as 250-255 to account for shadows. If it is observed that the flag color has dimmed, all that needs to be done is decrease these values.

- **Flag Parameters: Pop**

Represents the threshold area of the flag. It is recommended that this value be changed only with consultation.

- **Timeout:**

This value has no significance.