

**CONFIGURING VISION WORKSTATION  
CVS-1450 FOR ONLINE  
IC PIN COUNTING APPLICATION**

*A thesis*

*Submitted towards the partial fulfillment of requirements for the award of  
the degree of*

**Master of Engineering**

**in**

**Electronic Instrumentation & Control Engineering**

*Submitted by*

**Anurupa**

**Roll. No. 8044204**

*Under the Guidance of*

**Mr. Mandeep Singh**

**Assistant Professor**

**EIED**



**Department of Electrical and Instrumentation Engineering  
THAPAR INSTITUTE OF ENGINEERING & TECHNOLOGY  
DEEMED UNIVERSITY  
PATIALA – 147004**

**June 2006**

# Declaration

---

*I hereby declare that the report entitled “**Configuring Vision Workstation CVS-1450 for online IC Pin Counting Application**” is an authentic record of my own work carried out as requirements for the award of degree of M.E. (Electronic Instrumentation & Control) at Thapar Institute of Engineering & Technology (Deemed University), Patiala, under the guidance of Mr. Mandeep Singh (AP, EIED) during January to June 2006.*

**ANURUPA**

**Date: -----**

**Roll. No. 8044204**

*It is certified that the above statement made by the student is correct to the best of my knowledge and belief.*

(-----)

**Mr. Mandeep Singh**  
*Assistant Professor, EIED  
TIET, Patiala*

(-----)

**Mrs. Manbir Kaur**  
*Assistant Professor & Head EIED  
TIET, Patiala*

(-----)

**Dr. T.P. Singh**  
*Dean of Academic Affairs  
TIET, Patiala*

# Acknowledgment

---

*The real spirit of achieving a goal is through the way of excellence and austere discipline. I would have never succeeded in completing my task without the cooperation, encouragement and help provided to me by various personalities.*

*With deep sense of gratitude I express my sincere thanks to my esteemed and worthy supervisor, Mr. Mandeep Singh, Assistant Professor, Department of Electrical & Instrumentation Engineering, TIET (Deemed University) Patiala, for his valuable guidance in carrying out this work under his effective supervision, encouragement, enlightenment and cooperation.*

*I shall be failing in my duties if I do not express my deep sense of gratitude towards Mrs. Manbir Kaur, A.P. & Head of the Department of Electrical & Instrumentation Engineering, TIET (Deemed University) Patiala, who has been a constant source of inspiration for me throughout this work.*

*I am also thankful to all the staff members of the Department for their full cooperation and help.*

*Acknowledgement is due, to UGC Assistance for Strengthening Virtual Instrumentation as Special Paper in M.E. at Department of Electrical and Instrumentation, Thapar Institute of Engineering and Technology, Patiala under innovative program of teaching and research in inter-disciplinary and emerging area, for providing the financial help for purchase of the necessary equipment and software.*

*My greatest thanks are to all who wished me success especially my parents. Above all I render my gratitude to the ALMIGHTY who bestowed self-confidence, ability and strength in me to complete this work.*

*Place: TIET, Patiala*

*Anurupa*

*Date:*

# Abstract

---

*Machine Vision is an emerging area related to real-time capturing, processing, and analyzing the images for various kinds of scientific and industrial applications. Cell counting is required in number of applications in the fields such as Biotechnology, Material Science, Pathology, and Nano-technology etc. Automatic cell counting device has been designed using principle of Machine Vision to make the process easy, fast and accurate. Manual counting of large number of cell in any of these applications can be a tedious and time consuming process, prone to human errors. Keeping in view of this application of Machine Vision in medical field, we have used machine vision system, which categorize IC package by counting its number of pins and can be used in Electronic Industry. In the present work the image is captured by IEEE-1394 Digital Camera Prosilica 2.0.1 and processed by vision workstation CVS-1450. Configuring the two and optimizing their performance for the above application is one of the major components of this work.*

# Table of contents

---

<i>Declaration</i>	<i>i</i>
<i>Acknowledgment</i>	<i>ii</i>
<i>Abstract</i>	<i>iii</i>
<i>Table of contents</i>	<i>iv</i>
<i>List of figures</i>	<i>vii</i>
<i>List of tables</i>	<i>ix</i>
<i>List of abbreviations</i>	<i>x</i>
<b>CHAPTER 1</b>	<b>1</b>
<b>Introduction to Machine Vision basics</b>	<b>1-13</b>
<b>1.1 Introduction</b>	<b>1</b>
<b>1.2 Components of Machine Vision</b>	<b>1</b>
<b>1.3 Advantages and Limitations</b>	<b>2</b>
<b>1.4 Digital image basics</b>	<b>3</b>
<b>1.4.1 Digital image</b>	<b>3</b>
<b>1.4.2 Properties of a digitized image</b>	<b>4</b>
1.4.2.1 Image resolution	4
1.4.2.2 Image definition	4
1.4.2.2 Number of planes	4
<b>1.4.3 Image types</b>	<b>5</b>
1.4.3.1 Grayscale images	5
1.4.3.2 Color images	5
1.4.3.3 Complex images	5
<b>1.4.4 Image files</b>	<b>6</b>
<b>1.4.5 Region of interest</b>	<b>7</b>
<b>1.5 Applications of Machine Vision systems</b>	<b>7</b>
<b>1.5.1 Underwater vehicle</b>	<b>7</b>
<b>1.5.2 Acrylic fiber production</b>	<b>9</b>
<b>1.5.3 Agricultural robots</b>	<b>9</b>
<b>1.5.4 Smart sprayer</b>	<b>10</b>
<b>1.5.5 Inverted pendulum</b>	<b>10</b>
<b>1.5.6 Vehicle trajectory approximation</b>	<b>12</b>
<b>1.5.7 Fire detection in aircraft dry bays</b>	<b>13</b>
<b>CHAPTER 2</b>	<b>15-28</b>
<b>Requisite hardware</b>	<b>15</b>
<b>2.1 System requirements and installation</b>	<b>15</b>
<b>2.1.1 Installing Vision Assistant</b>	<b>15</b>
<b>2.1.2 Features</b>	<b>15</b>
<b>2.2 IMAQ for IEEE 1394 compatible cameras</b>	<b>16</b>

2.2.1 High-level functions	17
2.2.2 Low-level functions	17
<b>2.3 Acquisition flow</b>	<b>18</b>
<b>2.3.1 Initialization</b>	18
➤ Camera name	18
➤ Camera control mode	19
<b>2.3.2 Configuration</b>	19
➤ One-shot/Continuous acquisition	20
➤ Continues acquisition	20
➤ Number of buffers	20
➤ Region of interest	20
<b>2.3.3 Acquisition</b>	21
➤ User buffer	21
➤ Overwrite mode	21
➤ Timeouts	22
➤ Decoding	22
<b>2.4 Custom configuration</b>	<b>23</b>
<b>2.4.1 Installation of software package</b>	23
<b>2.4.2 Functions to be installed</b>	23
<b>2.5 CVS-1450</b>	<b>24</b>
<b>2.6 Hardware overview</b>	<b>24</b>
<b>2.6.1 Connections</b>	25
<b>2.6.2 LEDs, DIP switches, and connectors</b>	26
2.6.2.1 LEDs	26
2.6.2.2 Switches	27
2.6.2.3 Connectors	28
<b>CHAPTER 3</b>	<b>29-65</b>
<b>Configuration &amp; application software development</b>	<b>29</b>
<b>3.1 Introduction to Vision Assistant 7.1</b>	<b>29</b>
<b>3.2 Problem definition</b>	<b>29</b>
<b>3.3 Problem solution and its algorithm</b>	<b>30</b>
<b>3.3.1 Configuring the system</b>	30
3.3.1.1 Configuring CVS-1450	30
3.3.1.2 Configuring Prosilica camera 2.0.1	30
<b>3.3.2 Image acquisition</b>	31
<b>3.3.3 Process image</b>	32
➤ Acquire Image	32
➤ Extract color plane	32
➤ Image masking	32
➤ Threshold	32
➤ Morphological operations	32
➤ Particle analysis	32
<b>3.4 Software development in vision Assistant 7.1</b>	<b>32</b>
<b>3.4.1 Pin counting (40 pin IC) in vertical direction</b>	32
3.4.1.1 Welcome screen	32
3.4.1.2 Image acquisition	33
3.4.1.3 Extract color plane	34

3.4.1.4 Image masking	34
3.4.1.5 Store image	37
3.4.1.6 Image masking	37
3.4.1.7 Threshold	38
3.4.1.8 Erosion	39
3.4.1.9 Dilation	41
3.4.1.10 Particle filter	42
3.4.1.11 Particle analysis	43
3.4.1.12 Result	45
3.4.1.13 Estimating processing time	46
3.4.1.14 Performance meter	47
<b>3.4.2 Pin counting (40 pin IC) in horizontal direction</b>	<b>48</b>
3.4.2.1 Acquire image	48
3.4.2.2 Particle analysis	48
3.4.2.3 Results	49
<b>3.4.3 Pin counting (62 pin) in horizontal direction</b>	<b>50</b>
3.4.3.1 Acquire image	50
3.4.3.2 Results	50
<b>3.4.4 Pin counting (8 pin IC in vertical direction)</b>	<b>52</b>
3.4.4.1 Acquire image	52
3.4.4.2 Particle analysis	52
3.4.4.3 Results	53
<b>3.4.5 Pin counting (16 pin IC in inclined position)</b>	<b>53</b>
3.4.5.1 Acquire image	53
3.4.5.2 Particle analysis	54
3.4.5.3 Results	54
<b>3.4.6 Pin Counting (IC Base in inclined Position)</b>	<b>55</b>
3.4.6.1 Acquire image	55
3.4.6.2 Image masking	55
3.4.6.3 Threshold	56
3.4.6.4 Particle analysis	56
3.4.6.5 Result	57
3.4.6.6 Result 28 Pins	57
3.4.6.7 Result 80 Pins	58
3.4.6.8 Results 60 Pins	60
<b>3.5 Linking with LabVIEW</b>	<b>61</b>
3.5.1 Front panel	62
3.5.2 Block diagram	63
<b>3.6 Results and discussion</b>	<b>65</b>
<b>CHAPTER 4</b>	<b>66</b>
<b>Conclusions &amp; future scope</b>	<b>66</b>
<b>References</b>	<b>67</b>
<b>Publications from this research work</b>	<b>69</b>

# List of figures

---

	Page no.
Fig.1.1 Spatial reference of the (0, 0) Pixel	3
Fig.1.2 Hydrophone location and reception region	8
Fig.1.3 Intelligent sampling used in pattern matching	11
(a) Reference pattern	
(b) Sampled pattern spatial reference of the (0, 0) pixel	
Fig.1.4 Edge detection used in pattern matching	12
(a)Reference pattern	
(b) Edge information in the image	
Fig.1.5 (a) Landmarks to determine position and angle of the pendulum	12
(b) An image with the results obtained with pattern matching	
Fig.1.6 A left turn	13
Fig.2.1 Relationships between cameras, interface and camera files	19
Fig.2.2 CVS-1450 Series front panel	24
Fig.2.3 Hardware connections	26
Fig.2.4 Power OK and status LEDs	27
Fig.2.5 ACT/LINK LED and 100 Mbps LED	27
Fig.2.6 Safe Mode switch in on position	28
Fig.3.1 Acquiring images in Vision Assistant	33
Fig.3.2 Image acquisition	33
Fig.3.3 Extracting color plane	34
Fig.3.4 Image masking	35
Fig.3.5 Effect of an image masking	35
Fig.3.6 Selecting ROI	36
Fig.3.7 Effect of applying a mask with different offsets	36
Fig.3.8 Store an image in buffer	37
Fig.3.9 Image masking	37
Fig.3.10 Effect of threshold	38
Fig.3.11 Applying threshold	39
Fig.3.12 Effects of erosion	39

Fig.3.13 Erosion	41
fig.3.14 Effects of dilation	41
Fig.3.15 Dilation	42
Fig. 3.16 Selection of area of an image to analyze	44
Fig.3.17 particle analysis	45
Fig.3.18 Estimated times	47
Fig.3.19 Performance meter	47
Fig.3.20 Acquire image (40 pin IC horizontal)	48
Fig.3.21 Particle analysis	48
Fig.3.22 Acquire image	50
Fig.3.23 Acquire image (8 pin IC)	52
Fig.3.24 Particle analysis	52
Fig.3.25 Acquire image (16 pin IC)	53
Fig.3.26 Particle analysis	54
Fig.3.27 Acquire image (14 pin IC base)	55
Fig.3.28 Image masking	55
Fig.3.29 Threshold	56
Fig.3.30 Particle analysis	56
Fig.3.31 Front panel view	62
Fig.3.32 Block diagram part 1	63
Fig.3.33 Block diagram part 2	64
Fig.3.34 Block diagram part 3	64

# List of tables

---

Table 1.1 Bytes per pixel	6
Table 2.1 Camera naming convention	18
Table 2.2 Decoder inputs and corresponding outputs	23
Table 2.3 CVS-1450 device connectors	28
Table 3.1 Factor	42
Table 3.2 Particle measurements	44
Table 3.3 Pin count of 40 pin IC (Vertical)	46
Table 3.4 Pin count of 40 pin IC (Horizontal)	49
Table 3.5 Pin count of 62 pin device (Horizontal)	51
Table 3.6 Pin count of 8 pin IC (Vertical)	53
Table 3.7 Pin count of 16 pin IC (Inclined)	54
Table 3.8 Pin count of 14 pin IC	57
Table 3.9 Pin count of 28 pin IC	58
Table 3.10 Pin count of 80 pin IC	60
Table 3.11 Pin count of 60 pin IC	61

# List of abbreviations

---

- NI National Instruments
- IMAQ Image Acquisition
- VA Vision Assistant
- CVS Compact Vision System
- CCD Charge Coupled Devices
- LabVIEW Laboratory Instruments Engineers Workbench
- ROI Region of Interest
- RGB Red Green Blue
- HSL Hue Saturation Luminescence
- LED Light Emitting Diode
- IEEE Institute of Electrical and Electronics Engineers
- TTL Transistor -Transistor Logic
- Mbps Mega Bytes per Second
- Kbps Kilo Bytes per Second
- RS Recommended Standards
- RJ Registered Jack
- YUV Yellow Ultra violet
- DCAM Digital Camera

# Chapter 1

## Introduction to Machine Vision basics

---

### 1.1 Introduction

We see the world around through our eyes. Our eyes are the sensory organs that capture images and transmit to our brain at very fast rate. The image is representation of real scene either in black & white or in color. The brain performs various processing functions and vision is perceived. In human beings we make use of vision for accomplishing majority of our tasks. Blindfolding ourselves and observing how our daily routine is seriously hampered without our vision can easily verify this fact.

Although the first machine that captured image was a pinhole camera that was invented way back in 1850s, which was followed by many advances in image capturing techniques. Black & white camera gave way to colored camera, resolution of picture captured enhanced, moving pictures were captured using monochrome T.V. Camera followed by colored T.V. camera and now a days we have digital cameras as small as a size of button, embedded in our mobile phones, at a price, a student can afford from his pocket money.

However, when we talk of machine vision, these cameras alone do not “see” the way we do. With human vision the process of “seeing” i.e. capturing images takes place at a very fast rate, almost continuously for several hours. The most important aspect is equally fast processing of these images so as to accomplish necessary decisions and actions. Image processing is collection of programs and techniques that improve, simplify, enhance or otherwise alter an image. Machine vision also aims at aping this task of capturing images at fast rates and simultaneously processing and analysis of these images so that machine can “see”, the way we do.

### 1.2 Components of Machine Vision

Machine Vision is related to artificial intelligence. We take the images in real time and process them for extracting useful information from them. The system making use of machine vision has basically three components: Camera (Prosilica 2.0.1), Image Processor (CVS-1450), Display Unit (PC monitor).

**Camera** is used for taking pictures of desired field, spot, point or any moving object. Camera must have high grabbing or sampling rate (Frames to be taken in one second). Frames are to be taken at fast rate such that we do not lose information about the object. These frames are to be taken in real time that's why frames are blurred and this needs image processing. Cameras used are mainly CCD (Charge coupled Devices) type or digital cameras. Resolution of camera must be strictly decided according to the object's area and size [1]. In our work we have used digital camera PROSILICA 2.0.1, IEEE-1394 compatible.

**Image processor** is used to get the best out of grabbed or captured images we make use of image processor. Image processor is usually a computer, laptop or a digital signal processor. Image can also be processed using tools available in LabVIEW. Image is divided into small sections on screen called picture cells, or pixels, where the size of all pixels is the same, while intensity of light in each pixel is varied to create the images. Image processor processes the blurred image pixel by pixel or sometimes reference points are processed which represent the whole image and make the picture more clearly to the operator for further analysis. The more the pixels present, the better the resolution of camera. We have used Machine Vision Assistant 7.1 simulator software to perform image-processing tasks with CVS-1450. Basically VA 7.1 helps us to make LabVIEW program.

**Display unit** is mainly a screen having pixels to display the grabbed images. A personal computer or laptop is used for this purpose. Sometimes storage of grabbed images is also done with computer for offline analysis. We have used simple PC monitor to get images from camera. Camera is connected to monitor through CVS-1450 using LAN, which directly displays the acquired image on monitor.

### **1.3 Advantages and Limitations**

1. Machine vision can be implemented to various industrial locations where cameras can replace two or three workmen.
2. It can also be employed in hazardous conditions where it is very difficult for operator to watch the process disturbances continuously.
3. The decisions are taken by processor in real time therefore it is very useful in real time process control.

4. The grabbing rate of some specialized cameras can be too high as compared to human perception therefore it can detect very fast and minute changes in case moving object.
5. System performance is affected due to vibration and noise because camera produces blurred image in these conditions.
6. Natural light affect the camera performance due to changing of brightness of natural light from day to night.
7. Sometimes the images that are heavily distorted may loose useful information during various processing functions.
8. The accuracy of decision made by processor is simply dependent upon the resolution of camera. The more the resolution the finer will be the image.

## 1.4 Digital image basics

Digital Image Processing is done to process any image to get useful information from it. For Image processing, we must know about the image basics. These basics are given below:

### 1.4.1 Digital image

An image is a 2D array of values representing light intensity. For the purposes of image processing, the term image refers to a digital image. An image is a function of the light intensity  $f(x, y)$  where  $f$  is the brightness of the point  $(x, y)$ , and  $x$  and  $y$  represents the spatial coordinates of a picture element, or pixel. By convention, the spatial reference of the pixel with the coordinates  $(0, 0)$  is located at the top, left corner of the image. Notice in Figure 1.1 that the value of  $x$  increases moving from left to right, and the value of  $y$  increases from top to bottom.

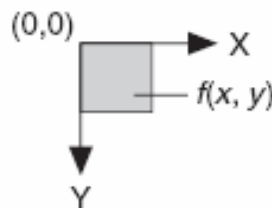


Figure 1.1 Spatial reference of the  $(0, 0)$  pixel

In digital image processing, an imaging sensor converts an image into a discrete number of pixels. The imaging sensor assigns to each pixel a numeric location and a gray level or color value that specifies the brightness or color of the pixel.

#### **1.4.2 Properties of a digitized image**

A digitized image has three basic properties: resolution, definition, and number of planes. These three properties are described.

##### **1.4.2.1 Image resolution**

Image Resolution is spatial resolution of an image, which is determined by its number of rows and columns of pixels. An image composed of  $m$  columns and  $n$  rows has a resolution of  $m \times n$ . This image has  $m$  pixels along its horizontal axis and  $n$  pixels along its vertical axis.

##### **1.4.2.2 Image definition**

Image definition indicates the number of shades that we can see in the image. The bit depth of an image is the number of bits used to encode the value of a pixel. For a given bit depth of  $n$ , the image has an image definition of  $2^n$ , meaning a pixel can have  $2^n$  different values. For example, if  $n$  equals 8 bits, a pixel can have 256 different values ranging from 0 to 255. If  $n$  equals 16 bits, a pixel can have 65,536 different values ranging from 0 to 65,535 or from  $-32,768$  to  $32,767$  as per manual [9]. The manner in which we encode our image depends on the nature of the image acquisition device, the type of image processing need to use, and the type of analysis we need to perform. For example, 8-bit encoding is sufficient if we need to obtain the shape information of objects in an image. However, if we need to precisely measure the light intensity of an image or region in an image, we must use 16-bit or floating-point encoding. Software does not directly support other types of image encoding, particularly images encoded as 1-bit, 2-bit, or 4-bit images. In these cases, Software automatically transforms the image into an 8-bit image the minimum bit depth for Software when opening the image file.

##### **1.4.2.2 Number of planes**

Number of planes corresponds to the number of arrays of pixels that compose the image. A grayscale or pseudo-color image is composed of one plane; while a true-color image is composed of three planes one each for the red component, blue component, and green component. In true-color images, the color component intensities of a pixel are coded into three different values. A color image is the combination of three arrays of pixels corresponding to the red, green, and blue components in a Red Green Blue (RGB) image.

Hue Saturation Luminescence (HSL) images are defined by their hue, saturation, and luminance values.

### **1.4.3 Image types**

There are three types of images: grayscale, color, and complex images. For an identical spatial resolution, a color image occupies four times the memory space of an 8-bit grayscale image, and a complex image occupies eight times the memory of an 8-bit grayscale image.

#### **1.4.3.1 Grayscale images**

Grayscale images are composed of a single plane of pixels. Each pixel is encoded using one of the following single numbers:

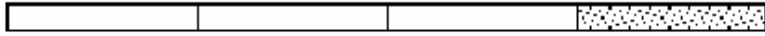
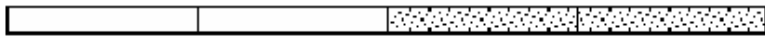


- An 8-bit unsigned integer representing grayscale values between 0 and 255
- A 16-bit signed integer representing grayscale values between  $-32,768$  and  $+32,767$
- A single-precision floating-point number, encoded using four bytes, represents grayscale values ranging from  $-\infty$  to  $\infty$ .

#### **1.4.3.2 Color images**

Color images are encoded in memory as a red, green, and blue (RGB) image. Color image pixels are a composite of four values. RGB images store color information using 8 bits each for the red, green, and blue planes. RGB 64 bit images store color information using 16 bits each for the red, green, and blue planes. In the RGB and HSL color models, an additional 8-bit value goes unused. This representation is known as 4- $\times$  8-bit or 32-bit encoding. In the RGB 64 bit color model, an additional 16-bit value goes unused. This representation is known as 4- $\times$  16-bit or 64-bit encoding.

#### **1.4.3.3 Complex images**

Complex image contains the frequency information of a grayscale image. We can create a complex image by applying a Fast Fourier transform (FFT) to a grayscale image. After we transform a grayscale image into a complex image, we can perform frequency domain operations on the image. Each pixel in a complex image is encoded as two single-precision floating-point values, which represent the real and imaginary components of the complex pixel. We can extract the following four components from a complex image: the real part, imaginary part, magnitude, and phase. Table 1.1 shows arrangement of number of Bytes per Pixel.

Image Type	Number of Bytes per Pixel Data
<b>8-bit (Unsigned) Integer Grayscale</b>  (1 byte or 8-bit)	 <p style="text-align: center;">8-bit for the grayscale intensity</p>
<b>16-bit (Signed) Integer Grayscale</b>  (2 bytes or 16-bit)	 <p style="text-align: center;">16-bit for the grayscale intensity</p>
<b>32-bit Floating- Point Grayscale</b>  (4 bytes or 32-bit)	 <p style="text-align: center;">32-bit for the grayscale intensity</p>
<b>RGB Color</b>  (4 bytes or 32-bit)	 <p style="text-align: center;">           8-bit for the alpha value (not used)                8-bit for the red intensity                8-bit for the green intensity                8-bit for the blue intensity         </p>

*Table 1.1 Bytes per pixel*

#### 1.4.4 Image files

An image file is composed of a header followed by pixel values. Depending on the file format, the header contains image information about the horizontal and vertical resolution, pixel definition, and the original palette. Image files may also store information about calibration, pattern matching templates, and overlays. The following are common image file formats:

- Bitmap (BMP)
- Tagged image file format (TIFF)
- Portable network graphics (PNG) offers the capability of storing image information about spatial calibration, pattern matching templates, and overlays
- Joint Photographic Experts Group format (JPEG)

- National Instruments internal image file format (AIPD) Used for saving floating-point, complex, and HSL images.

Standard formats for 8-bit grayscale and RGB color images are BMP, TIFF, PNG, JPEG, and AIPD. Standard formats for 16-bit grayscale, 64-bit RGB, and complex images are PNG and AIPD

#### **1.4.5 Region of interest**

A region of interest (ROI) is an area of an image in which we want to perform our image analysis. We use ROIs to focus our processing and analysis on part of an image. We can define an ROI using standard contours, such as an oval or rectangle, or freehand contours also can perform any of the following options:

- Construct an ROI in an image display environment
- Associate an ROI with an image display environment
- Extract an ROI associated with an image display environment
- Erase the current ROI from an image display environment
- Transform an ROI into an image mask
- Transform an image mask into an ROI

### **1.5 Applications of Machine Vision systems**

Machine Vision has vast applications in many fields of science and Technology. Some of applications of Machine Vision System are:

#### **1.5.1 Underwater vehicle**

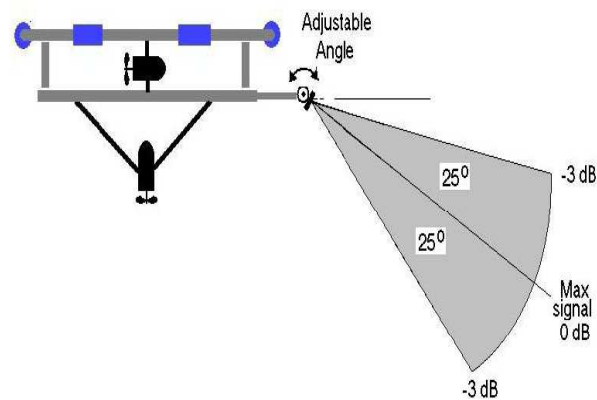
An autonomous underwater vehicle is a self-propelled, self-controlled machine and is capable of underwater motion with associated sensors and actuators. Objective of this vehicle is to search, locate, and identify a number of objects and determine their depth. David Eaton et al., [2] developed an underwater vehicle with the motors that allow motion along the X-direction (forward/backward), motion along Z-axis (lift/dive) and rotation about the Z-axis (yaw) for steering.

Activating the two side motors in the same direction activates forward and backward motion. On the spot rotation is possible by activating two sides' motors in the opposite directions at same power level. Buoyancy is also adjustable by attaching floatation devices or weights as needed to the frame. The control program is written in LabVIEW. Navigation control is based on an electronic compass that allows underwater vehicle to locate and track a given direction. The electronic compass provides a serial 8-

bit number that corresponds to a heading angle. The compass interfaces to the laptop and is processed through LabVIEW program to control and maintain vehicle motion according to a planned path. The weight distribution maintains minimal roll and pitch by providing pendulum action and stability to the vehicle. Pressure sensor allows underwater vehicle to measure its depth. Figure 1.2 shows the arrangement of vehicle.

The underwater vehicle controller activates the lift/dive motor to achieve a desired depth in a control loop using feedback from the depth sensor. Navigation control system relies on the compass and depth sensor to follow a pre-program initial search pattern. Two hydrophones located in front of the vehicle and spaced 25 inches apart form the ears of vehicle. They provide directional information for locating and approaching pinger (sound produced by creature).

Barcode detection is done with a camera used to provide images of the scenery below the submersible. The acquired images are processed in search of man-made object with a flat top surface having identifying barcodes. Each acquired image is scanned for the presence of a barcode. If a barcode is found, it is decoded to identify the object and the vehicle proceeds to determine the depth of the object. When barcodes are found, their depth is estimated and stored. The object identity and its measured depth are stored for easy retrieval upon return of the submersible to base.



*Figure 1.2 Hydrophone location and reception region*

In depth measurement, the submarine dives under the guidance of the vision system until the barcode images attains a given size on the image corresponding to a known predetermined distance between the camera and the object and estimate the object depth from the submersible depth sensor measurement.

### **1.5.2 Acrylic fiber production**

The production of acrylic fibers on the manufacturer's production line begins with extrusion of liquid polymer through spinnerets (thin metal diaphragms perforated by thousands of small holes) into high temperature water bath. The solidifying fiber forms six parallel, flattened bundles or tapes. The tapes are drawn through production line machinery where they are progressively stretched and dried over a series of heated metal cylinders. The width of the interval between them has an important influence on the quality of the fibers produced and so it is essential that these parameters be kept in strict tolerances. Paulo Heleno et al., [3] developed a system using machine vision with the following objectives:

- Perform continuous automatic inspection of production lines
- Detect anomalies in fiber tape width and spacing
- Detect gaps between the fibers that form each tape
- Detect fibers wrapped around the production line cylinders

The system consists of network of PCs monitoring each production line and all PCs are controlled centrally. Each PC is equipped with frame grabber. It performs automatic inspection through specially designed measuring algorithms using four video cameras located at key positions on production lines. The inspection is limited to small areas of the images, the inspection windows.

When running in automatic inspection mode, the system warns the user at the central station when defects are detected. Algorithms developed for this system include the fiber tape and wrap detection criteria, and the calibration algorithms. Fibers tape and wrap detection algorithm and edge detection algorithms are used to detect the wrap and gap between the tapes. These bands contrast with the background cylinder; meaning that fibers tape and wrap segmentation is essentially an edge detection and edge matching process. The images obtained at each inspection station will appear as vertical bands. Illumination conditions are also change throughout a 24 hours period, so the system is trained according to lighting conditions.

### **1.5.3 Agricultural robots**

In last few years, robotics has been increasingly adopted in agriculture. Computer vision is applied to agricultural robotics, which performs automated harvesting and post harvesting tasks, which are generally, time consuming, tiring and particularly demanding. Also harvest labor accounts for as much as one-half to two-thirds of the total labor costs.

Milella et al., [4] developed harvesting process of fennel (Saunf: Indian name) and post harvesting process of Radicchio (a vegetable grown widely in Italy) using machine vision. In case of post harvesting process of fennel, fennel is moved on test-bed comprising conveyer belt and a camera, which is mounted pointing straight down 1m across from the conveyer belt. The camera is attached to the end-effector of the cutting machine. The software flag detects when a plant enters the image. The useful portion of fennel is in between the root and the leaves. Camera grabs the picture and computer processes this image to remove disturbance with image processing programs stored in the computer. After this, image analysis is done to determine the useful portion of fennel. Cutting machine works on decisions made by computer.

Another application is harvesting of radicchio, a vision-based program is used aiming at localizing the plants in the fields. Once the radicchio is located, the target is set for the control system and the end-effectors for harvesting. It locates the vegetable among vegetable terrain (land); leaves and small rocks etc. Image Processing and analysis again helps in taking decisions.

#### **1.5.4 Smart sprayer**

Herbicides make up a large component of the pesticides used and are relied on heavily for effective weed control resulting in reduced yield loss in grain production. Also they represent a costly input to farmers and a source of ground water contamination. Typically, herbicides are applied with a blanket treatment to a whole field without regard to the spatial variability of the weeds in the field. This practice results in some areas where no or few weeds exist receiving just as much chemicals as those areas with high densities of weeds. If a more sophisticated chemical delivery system is developed which applied chemicals where weeds existed and abstained where there were no weeds, chemical usage would-be reduced and chemicals would be more effectively placed.

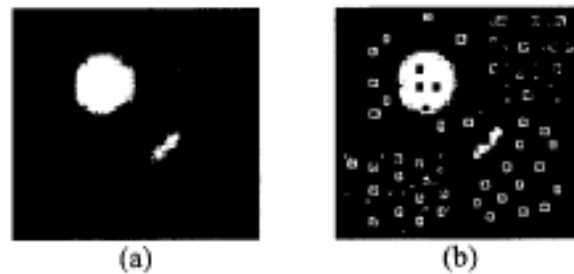
Steward et al., [5] developed a vision based spraying system with the objective to sense the weed in crop fields in variable lighting and control the dose of spray accordingly. Images of weeds are acquired between the rows of soybeans. An environmentally adaptive segmentation algorithm (EASA) is used to overcome light variation.

#### **1.5.5 Inverted pendulum**

The problem of control of inverted pendulum with feedback is traditionally tackled by taking signal from encoders to return the position of cart and angle of the rod.

Wenzel et al., [6] tried to solve this problem with a new approach using computer vision system as visual encoders. Determination of cart position and angle of the rod are based on pattern matching algorithms using landmarks. Two landmarks (templates) are placed on the top and the base of the pendulum.

The location of the landmark at the base of the pendulum is used to calculate the position of the cart at any given time. The angle of the pendulum is determined by using the location of both the landmarks. A high sampling rate is obtained by using a Kodak ES-310 digital camera delivering 85 frames per second with a good resolution: 640 x 480 pixels non-interlaced. As the pendulum is in constant motion, the exposure time of the camera is decreased as much as possible in order to reduce the blur in the acquired image. The programming of the underlying control system is completely graphically oriented using a control and simulation package. The complete solution is based on the graphical language LAB VIEW. Pattern matching is the technique used in machine vision to quickly locate known reference patterns in images 1.3(a) & 1.3(b). With pattern matching a model or template that represents the object is created.



*Figure 1.3 Intelligent sampling used in pattern matching (a) Reference pattern, (b) Sampled pattern*

Pattern matching can provide information about the presence or absence, number, and location of the model within an image. By using all the information present in the image to pattern matching makes the process time intensive and less accurate. Therefore by sampling the image and extracting a fewer number of points that represent overall content of the image, the speed as well as the accuracy of the pattern matching tool can be improved. The pattern-matching tool also utilizes the edge information in the image to provide information about the structure of the image. The edge image also contains a reduced amount of very significant data. Pattern matching then reduces to matching the

edge and/or geometric information between the template and the image shown in figure below:



*Figure 1.4 Edge detection used in pattern matching. (a) Reference pattern (b) Edge information in the image.*

Pattern matching is used to determine the location of center of landmarks in each acquired image. The location of the landmark at the base of the pendulum is used to calculate the position of the cart at any given time. The angle of the pendulum is determined by using the location of both the landmarks.



*Figure 1.5 (a) Landmarks used to determine position and angle of the pendulum. (b) An image with the results obtained with pattern matching*

### **1.5.6 Vehicle trajectory approximation**

As the density of road traffic increases it becomes ever more important to detect quickly accidents or other abnormal events, both to save lives and to reduce the disruptive effects on traffic flow. Certain events can already be detected automatically using the image sequences obtained by fixed surveillance cameras, which already line many motorways and main roads. 2D tracking is not sufficient to recover the more detailed information needed for accurate monitoring and control of road traffic. This information can only be obtained by taking into account the 3D nature of the scene. Fraile et al., [7] developed a technique for finding low curvature smooth approximations to trajectories in the plane. The method is applied to short segments of a vehicle trajectory in a known

ground plane. Estimates of the speed and steering angle are obtained for each segment and the motion during the segment is assigned to one of the four classes: ahead, left, right, stop. A hidden Markov model for the motion of the car is constructed and the Viterbi algorithm is used to find the sequence of internal states for which the observed behavior of the vehicle has the highest probability. The model contains four states which are in order A, L, R, S. These are the true states of the car, corresponding to ahead, turning left, and turning right, stopped. They are to be inferred from the measurements. The HMM has four output symbols in order a, l, r, s.

To start with, a sequence of measurements is obtained; then the trajectory of the vehicle is classified. The measurement sequence is divided into overlapping segments. In each segment the trajectory of the car is approximated by a smooth function and then assigned to one of four categories: ahead, left, right or stop such that set {a, l, r, s} where each letter corresponds to the appropriate category. The sequence of measurements for each trajectory is divided into overlapping segments, each containing 10 measurements. Each pair of adjacent segments overlapped by 9 measurements.

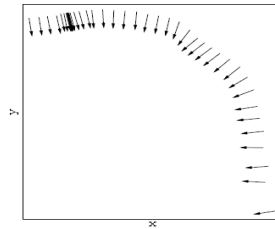


Figure 1.6 A left turn

The car moves from bottom right to top left Figure1.6. The arrow represents the normal to the trajectory, drawn as if the driver has ending the right arm out of the window. The base points of the arrows are the coordinates of the vehicle on each sample. The sequence of extracted symbols and the most likely sequence of states, as identified by the algorithm, are

```
a a a a a a a l l l l l a a
a l l a a l l l
A A A A A A A L L L L L L L
L L L L L L L L
```

### 1.5.7 Fire detection in aircraft dry bays

A machine vision approach can be applied to detect hydrocarbon fires in aircraft dry bays and engine compartments. Projectiles may cause electrical sparks etc resulting in

fires on-board military aircraft. The combination of heat, fumes and oil from hydraulics, fuel lines, etc. can account of most of the false alarms or failures of conventional smoke/heat sensors as they have too slow response times. Simon Y. Foo [8] developed a visible spectrum machine vision system to detect and characterize hydrocarbon fires. In this system visible characteristics of fires such as brightness, color, spectral flicker, and stationarity, are used to discriminate from other visible stimuli and background. To detect and characterize a fire in terms of growth pattern is very important feature of a fire alarm system. The fire alarm must operate in real time as well as minimize false alarms due to artificial sources such as flashlights and halogen lights.

The frames are to be taken at very fast rate, at about 30 frames per second. If each frame from a video camera is compared with previous frame, any movement or changes within the field of view of camera can be detected. Subtracting perfectly aligned pixels of two sequential videos images, followed by a threshold process and finally a pixel difference tally, does this. If the tally of pixels that differ in two sequential video images exceeds a predetermined threshold, then substantial motion or changes are detected. In a similar manner, the ignition and growth of fire can be detected by the subtraction of two sequential images. The inputs to the machine vision system consists of set of statistical measures derive from the histogram and image subtraction analysis of successive image frames.

# Chapter 2

## Requisite hardware

---

### 2.1 System requirements and installation

To run Vision Assistant, system must meet the following minimum requirements

- Personal computer using a 233 MHz Pentium-class processor. Using a Pentium III or Celeron 600 MHz or equivalent is recommended.
- Microsoft Windows 2000/NT/XP. If we are using Windows NT 4.0, we must have Service Pack 6 or later installed to run Vision Assistant.
- 1024 , 768 resolution or higher video adapter; 65,536 colors, 16-bit or higher.
- Minimum of 128 MB RAM; 256 MB recommended.
- Minimum of 200 MB of free hard disk space.
- If we are acquiring images, the system must have National Instruments image acquisition (IMAQ) hardware and NI-IMAQ 3.0 or later or NI-IMAQ for IEEE 1394 Cameras 1.5 or later installed.

#### 2.1.1 Installing Vision Assistant

- To install Vision Assistant on a Windows 2000/NT/XP system, we must be logged in with Administrator privileges.
- Insert the Vision Assistant CD into the CD-ROM drive.
- If we do not have auto run enabled, double-click `autorun.exe`. If we have auto run enabled, `autorun.exe` runs automatically.

#### 2.1.2 Features

Vision Assistant offers the following features:

- **Script window:** It Records a series of image processing steps and the settings we use for each of those steps. We can run scripts on single images or in a batch to analyze a collection of images. We also can modify and save scripts.
- **Image browser:** It contains all of the images currently loaded in Vision Assistant. We can select an image to process by double-clicking it in the Image Browser.

- **Processing window:** It updates the image as we change parameters. Because this view immediately reflects the changes we have made in the Parameter window, we can continue modifying parameters until we get the result we want.
- **Functions window/Parameter window:** It displays a list of image processing functions we can use to develop an algorithm, or displays parameters that we can set for an image processing function. Each function available through the Functions window has a Parameter window in which we set the parameters for that function.
- **Reference window/Embedded Help window:** The Image tab of the Reference window displays the image source as we manipulate it in the Processing window. The other tabs in the Embedded Help window contain context help for the function we are using.
- **Solution wizard:** It displays a list of industries and corresponding quality-assurance tasks that those industries perform. The wizard loads an IMAQ Vision-based solution for the task we select.
- **Performance meter:** It estimates how long a script will take to complete on a given image.
- **LabVIEW VI creation:** It creates a LabVIEW VI corresponding to the algorithm we prototype in Vision Assistant. Based on the options we select, the LabVIEW VI Creation Wizard creates a new VI that implements the image processing steps of the current script or of a saved script file.
- **C Code creation:** It creates a C file corresponding to the algorithm we prototype in Vision Assistant. Based on the options we select, the C Code Creation Wizard creates a C function that implements the image processing steps of the current script.
- **Builder file:** ASCII text file that lists the C and Microsoft Visual Basic functions and parameters for the algorithm we prototyped in Vision Assistant.

## 2.2 IMAQ for IEEE 1394 (Fire wire) compatible cameras

The NI-IMAQ for IEEE 1394 compatible cameras software, lists the supported application development environments describes the fundamentals of creating

applications using NI-IMAQ for IEEE 1394 Cameras, describes the files used to build these applications. NI-IMAQ Software for IEEE 1394 Cameras gives us the ability to use IEEE-1394 industrial digital video cameras to acquire images. The cameras may operate at various resolutions and frame rates, depending on camera capabilities.

We use National Instruments Measurement & Automation Explorer (MAX) to configure our IEEE-1394 camera and Refer to the NI-IMAQ for IEEE 1394 Cameras help for information about configuring our IEEE-1394 camera. The camera configuration is saved in a camera file, which the NI-IMAQ for IEEE-1394 Cameras VIs and functions use to configure a camera and supported attributes. The NI-IMAQ for IEEE-1394 cameras application programming interface (API) is divided two main function groups: high-level and low-level.

### **2.2.1 High-level functions**

Use to capture images quickly and easily. If we need more advanced functionality, we can mix high-level functions with low-level functions.

- **Snap functions**

Capture all or a portion of a single image to the user buffer.

- **Grab functions**

Perform an acquisition that loops continually on one or more internal buffers. We can copy the last acquired buffer to a separate user buffer for processing or analysis.

- **Sequence functions**

Acquire a specified number of internal buffers and then stops.

- **Trigger functions**

Control the trigger mode of the IEEE-1394 camera.

### **2.2.2 Low-level functions**

Use when we require more direct control of the image acquisition.

- **Acquisition functions**

Configure, start, stop, and unconfigure an image acquisition, or examine a user buffer during an acquisition.

- **Attribute functions**

Examine and change the acquisition or camera attributes.

➤ **Utility functions**

Display an image in a window, save an image to a file, or to get detailed error information.

Both high-level and low-level functions support snaps, grab, sequence, and triggered acquisitions. Using high-level functions, we can write programs quickly without having to learn the details of the low-level API and driver. The low-level functions give us finer granularity and control over the image acquisition process, but we must understand the API and driver in greater detail to use these functions.

### 2.3 Acquisition flow

The basic steps of performing an acquisition with the NI-IMAQ for IEEE 1394 cameras software .The basic steps are initialization, configuration, and acquisition.

#### 2.3.1 Initialization

To acquire images using the high-level or low-level functions, we first must initialize a camera session. A camera session is a process-safe handle to an IEEE 1394 camera. The driver uses a camera session to identify the camera to which further NI-IMAQ for IEEE 1394 Cameras functions apply. We can simultaneously open as many camera sessions as there are cameras connected to our system. When initializing the camera session, we need to specify two parameters: Camera name and Camera control mode.

➤ **Camera name**

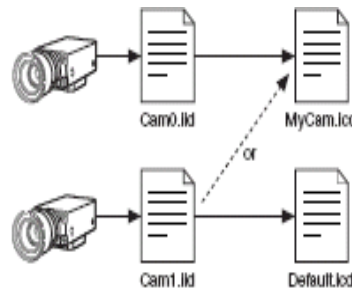
NI-IMAQ for IEEE 1394 Cameras references all camera sessions by a name. The driver creates default names for each camera in our system in the order that the cameras are connected. The names observe the convention shown in Table 2.1.

Camera Name	IEEE 1394 Camera Installed
Cam0	Device 1
Cam1	Device 2
Cam.....n	Device.....n

*Table 2.1 Camera naming conventions*

**Interface files** store information about which physical camera is associated with a camera name. Only a single camera can use each interface file.

**Camera files** store all the configurable attributes. Camera files can be shared between identical cameras. Use MAX to configure the default state of a particular camera. Figure 2.1 shows the relationship between cameras, interface files, and camera files.



*Figure 2.1 Relationships between cameras, interface files, and camera files*

Use the Enumerate function to query the number and names of available cameras. When we open a camera session with the Initialize function, the camera with the unique serial number described by the interface file camn.iid opens, where n is the reference to the camera. If the camera is not present and a camera of the same make and model is present, as described in the interface file, the driver opens the available camera. The interface file updates to use the new camera. The camera file described by the interface file opens, and all the user attributes are set in the driver. If no camera of the same make and model is present, the Initialize function returns an error.

#### ➤ **Camera control mode**

The camera control mode parameter has two options: controller and listener. The default option controller controls the camera and receives video data. The listener only receives video data. We use the listener option in broadcasting applications.

### **2.3.2 Configuration**

After initializing the interface, configure the interface for acquisition by specifying the following parameters: whether the acquisition is one-shot or continuous, the number of internal buffers to use, and the region of interest for the acquisition. During configuration, the driver validates all the user-configurable attributes. If any attributes are invalid or out of range, the driver returns an error and does not configure the acquisition. If we want to reconfigure the acquisition, call the Clear Acquisition function before calling the configure function again.

➤ **One-shot/Continuous acquisition**

We use a one-shot acquisition to start an acquisition, perform the acquisition, and stop the acquisition using a single function. The number of images acquired is equal to the number of images in the images collection. With a one-shot acquisition, we specify a certain number of internal buffers. The camera transfers each image up to and including the specified number of buffers. The driver acquires every image during a one-shot acquisition. National Instruments recommends one-shot acquisition for applications that do not require real-time acquisition or processing.

➤ **Continues acquisition**

We use a continuous acquisition to start an acquisition, continuously acquire images into the internal buffers, and explicitly stop the acquisition. With continuous acquisition, the driver acquires video data continuously from the camera and enables us to examine the most current buffer. National Instruments recommends continuous acquisition for real-time acquisition and processing.

➤ **Number of buffers**

Another aspect of configuration is specifying the number of internal buffers into which we want to acquire image data. During configuration, buffers are allocated from system memory and page-locked. Once the acquisition starts, the camera transfers video data over the IEEE-1394 bus to the IEEE-1394 interface card FIFO. Then, video data is directly transferred to the internal buffer. This transfer requires negligible CPU resources. Each internal buffer we allocate is the exact size of the raw data being transmitted by the camera. For continuous acquisitions, allocate three or more buffers. Allocating a single buffer for a continuous acquisition may result in a high number of lost images. For one-shot acquisitions, specify the number of buffers that the application requires. For example, if the application runs for two seconds, and the camera acquires at 30 frames per second, allocate 60 buffers to capture each image.

➤ **Region of interest**

The region of interest (ROI) specifies a rectangular portion of the image to be captured. In Partial Image Size Format (Format 7) video modes, the ROI defines the portion of the image to transfer from the camera to system memory. In non-Format 7 video modes, the entire image is transferred from the camera to system memory. In all video modes, the ROI specifies the amount of data decoded by the driver while acquiring

into a user buffer. By default, the driver transfers the entire image. Specify a smaller ROI for the following reasons:

- To acquire only the necessary subset of data
- To increase the acquisition speed by reducing the amount of data transferred and/or decoded
- To allow for multiple simultaneous acquisitions by reducing bandwidth usage

### 2.3.3 Acquisition

After configuring and starting our acquisition, the camera sends data to the internal buffers. To process the acquired image data, we must copy the data from the internal buffer into our user buffer.

#### ➤ User buffer

Before starting the acquisition, we must allocate a user buffer in addition to configuring internal buffers. The driver copies or decodes image data from the internal buffer into the user buffer during acquisition. Then, process and analyze the image in the user buffer. When acquiring data into an IMAQ Vision image, the driver resizes and casts the image as needed. However, if we acquire data into a user buffer, we must allocate enough space for one decoded image.

A buffer number is a zero-based index that represents the cumulated transferred image count. For example, during a continuous acquisition with three internal buffers, the buffer number is updated as follows: 0, 1, 2, 3, 4, 5, and so on. Buffer numbers 0 and 3 refer to the same internal buffer in the buffer ring. For a one-shot acquisition, we can request only one of the available buffer numbers. For a continuous acquisition, we can request any present or future buffer number [11]. We can also request the next logical buffer or the buffer containing the most recently acquired data. With high-level grab acquisitions, the buffer number defaults to the next transferred buffer. When we complete the buffer acquisition step, the driver returns the actual buffer number with the image.

#### ➤ Overwrite mode

A continuous acquisition acquires and processes every image that is transferred from the camera. However, because of processing time fluctuations, some images from the camera may not be processed before the camera transfers the next image. Using multiple internal buffers in a continuous acquisition allows for a small amount of jitter. However, if a delay becomes too long, the camera overwrites the requested buffer with new image

data. NI-IMAQ for IEEE-1394 Cameras is able to detect overwritten internal buffers. We can configure the driver to manage an overwritten buffer in one of the following ways:

- Get newest valid buffer
- Get oldest valid buffer
- Fail and return an error

In all cases, the camera continues to transfer data when a buffer is overwritten. The default overwrite mode for all types of acquisition is to get the newest valid buffer. This option, which National Instruments recommends for most applications, enables us to process the most recent image. If we need to get the image closest in time to a requested buffer, configure the driver to get the oldest valid buffer. If our application requires that every image be processed, configure the driver to fail when a buffer is overwritten so that we are alerted.

➤ **Timeouts**

A timeout is the length of time, in milliseconds, that the driver waits for an image from the camera before returning an error. A timeout error usually occurs if the camera has been removed from the system or when the camera did not receive an external trigger signal.

➤ **Decoding**

Except for 8-bit monochrome images, all video modes require decoding before we can interpret the image data. For example, many color IEEE 1394 cameras output images of type YUV 4:2:2. However, IMAQ Vision does not natively support the YUV mode. To process and display the image, the driver automatically decodes the YUV image into a 32-bit RGB image. Table 2.2 lists common video modes and their corresponding image types after being decoded by NI-IMAQ for IEEE 1394 Cameras.

Raw Camera Output	Decoded Destination Image
8-bit monochrome	8-bit monochrome
16-bit monochrome	16-bit monochrome
YUV 4:1:1	32-bit color
YUV 4:2:2	32-bit color
YUV 4:4:4	32-bit color
24-bit RGB	32-bit color
48-bit RGB	64-bit color
8-bit Bayer	32-bit color
16-bit Bayer	32-bit color

*Table 2.2 Decoder inputs and corresponding outputs*

Decoding images requires CPU resources. However, many of the decoding algorithms have been optimized in the driver. If we do not want decoded image data, we can use NI-IMAQ for IEEE 1394 Cameras to get a copy of the raw camera output.

## **2.4 Custom configuration**

For acquiring images in real time we have to configure system according to our requirements. There are total five CDs to be installed in system which is shown below. Drivers for Data Acquisition Instrument Control Motion and Vision are in 2 CDs and remaining soft wares have one CD only.

### **2.4.1 Installation of software package**

The software package includes the following:

- NI vision Development module
- Drivers for Data Acquisition Instrument Control motion and Vision
- IEEE-1394 PROSILICA camera 2.0.1
- Lab View Vision Assistant 7.1

### **2.4.2 Functions to be installed**

- NI-Motion
- NI-DAQ
- NI-VISA
- NI-Vision
- NI-License Manager
- NI-488.2

## 2.5 CVS-1450

NI CVS-1450 Series devices are easy-to-use, distributed, real-time imaging systems that acquire, process, and display images from IEEE-1394 cameras conforming to the 1394-based Digital Camera Specification, Version 2.0.1. The CVS-1450 series also provides multiple digital input/output (I/O) options for communicating with external devices to configure and start an inspection and to indicate results. An Ethernet connection between the CVS-1450 device and a development computer allows to display measurement results and status information and to configure the CVS-1450 device settings. When configured, the CVS-1450 device can run applications without a connection to the development computer.

## 2.6 Hardware overview

Components of system Hardware are:

- CVS-1450 device
- Ethernet
- Windows 2000/XP/Me/98
- DCAM, IEEE-1394 camera

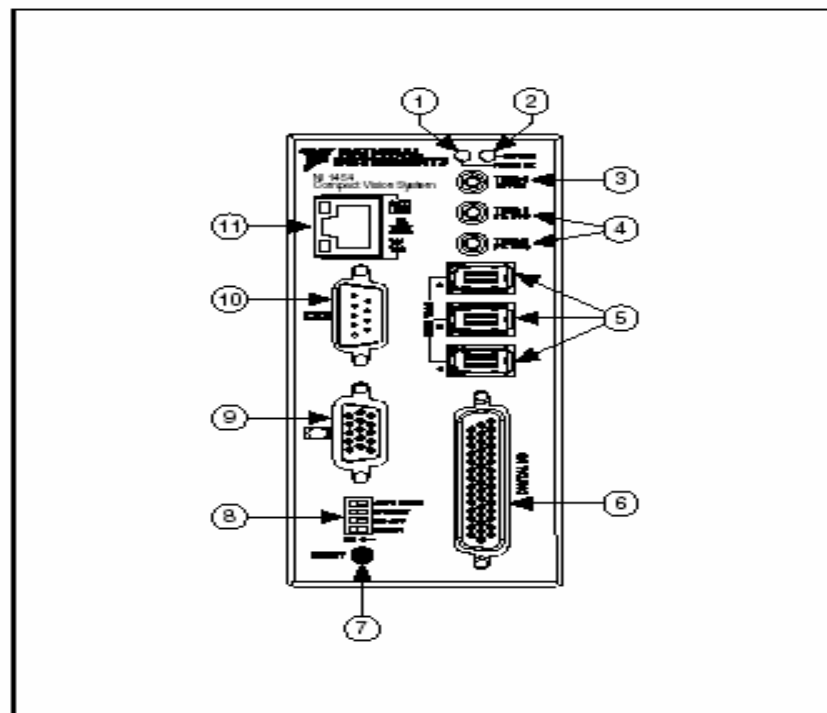


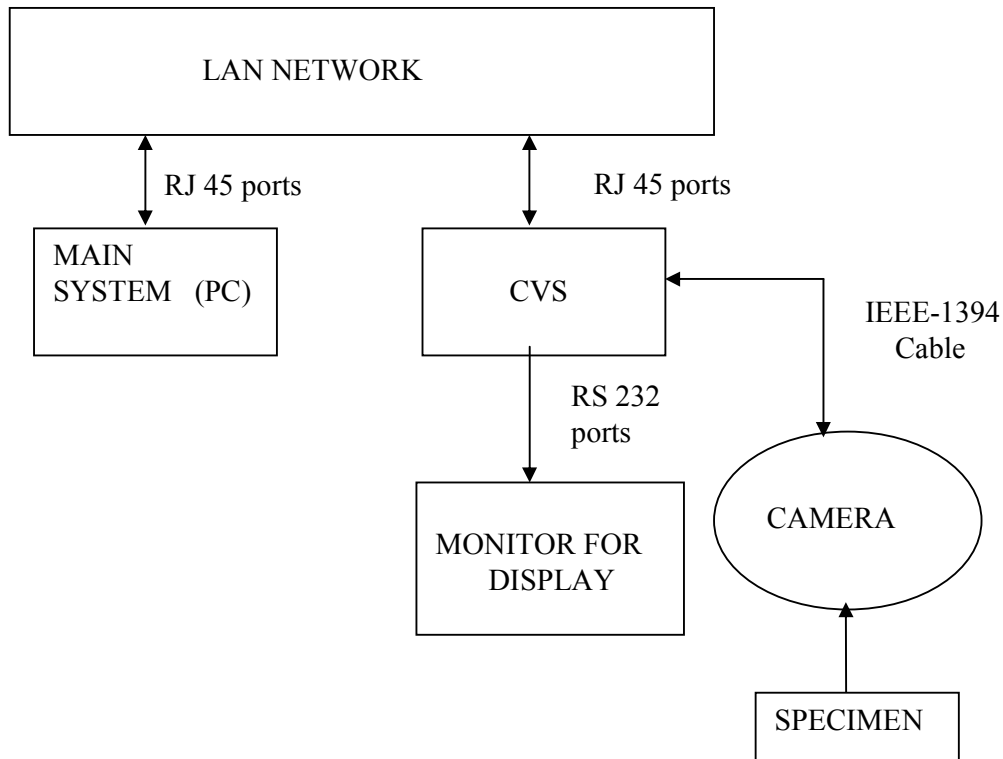
Figure 2.2 CVS-1450 series front panel

The CVS-1450 device front panel consists of:

1. Power LED
2. Status LED
3. Isolated Digital Input
4. TTL Digital Outputs
5. IEEE 1394 Ports
6. TTL I/O and Isolated I/O
7. Reset Button
8. DIP Switches
9. VGA
10. RS-232 Serial
11. RJ-45 Ethernet Port

### **2.6.1 Connections**

- We select the IP address of the main system (ISD-4) as 192.68.0.20.
- We have one monitor for taking display of real time image and it is connected directly to CVS and power supply.
- RS-232 Serial is used to attach the screen of the other PC on which the display is to be viewed.
- CVS is provided with power supply.
- The main system is connected to CVS through LAN therefore IP addresses must be compatible to each other.
- The RJ-45 Ethernet Port of main system which is provided for the LAN connections is connected to Lab LAN network.
- CVS RJ-45 Ethernet Port is also connected to LAN network and now both the devices are connected through LAN.
- The IEEE-1394 camera cable is used to make the connection between camera and CVS.
- One end is attached to the IEEE-1394 ports of CVS and other to the camera.



*Figure 2.3 Hardware connections*

## **2.6.2 LEDs, DIP switches, and connectors**

This section provides information about the location and functionality of the LED indicators, DIP switches, and connectors on the NI CVS-1450 device. The Connectors section provides signal names and descriptions for each connector.

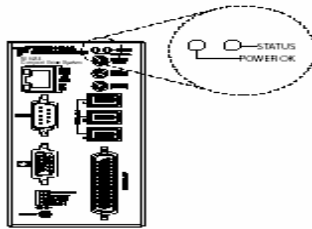
### **2.6.2.1 LEDs**

#### ➤ Power OK LED

Under normal operating conditions, the Power Ok LED remains green while the CVS-1450 device is powered on. A green Power Ok LED indicates that CVS-1450 device main power is receiving power and that the CVS-1450 device is not in a fault state. A red Power Ok LED indicates that the CVS-1450 device has shut down because of a fault state. A fault state occurs when the user shutdown input is asserted, the processor overheats, or the watchdog timer expires. The Power Ok LED does not indicate the status of the isolated power.

➤ Status LED

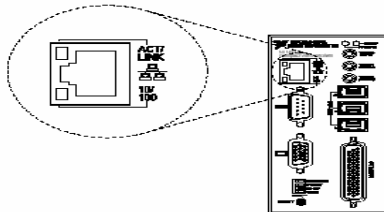
The orange Status LED remains off under normal operating conditions and flashes a specific number of times to indicate error conditions or certain DIP switch settings. The Status LED remains lit if the CVS-1450 device detects an internal error.



*Figure 2.4 Power OK and status LEDs*

➤ ACT/LINK LED

The orange ACT/LINK LED blinks when the CVS-1450 device receives data from or transmits data to the network through the Ethernet connection. Unrelated network activity causes this LED to blink occasionally even when the CVS-1450 device is inactive. Figure 2.5 shows the location of the ACT/LINK LED on the CVS-1450 device.



*Figure 2.5 ACT/LINK LED and 100 Mbps LED*

➤ 100 Mbps LED

The green 100 Mbps LED is lit when the network provides 100 Mbps support and the CVS-1450 device is communicating at 100 Mbps. If the 100 Mbps LED is not lit, the CVS-1450 device is not operating at 100 Mbps. Figure 2.4 shows the location of the 100 Mbps LED on the CVS-1450 device.

### **2.6.2.2 Switches**

➤ Safe mode switch

To start the CVS-1450 device in Safe mode, the safe mode switch is moved to the on position; it reset the CVS-1450 device. Safe mode is used to reconfigure TCP/IP settings and to download or update software from the development to resume normal operations; the CVS-1450 device is restarted with the Safe Mode switch in the off position.

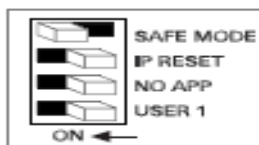


Figure 2.6 Safe mode switch in the on position

➤ IP Reset switch

To clear the CVS-1450 device IP settings, the IP Reset switch is moved to the ON position and CVS-1450 device is then reset. Resetting the CVS-1450 device with the IP RESET switch in the ON position resets the IP address to 0.0.0.0. A new network configuration for the CVS-1450 device from a development machine on the same subnet can be set up.

➤ NO APP switch

To prevent the CVS-1450 device from automatically running VIs at startup, the NO APP switch is moved to the ON position and after that CVS-1450 device is made to reset. If the CVS-1450 device becomes inaccessible because of a startup program, NO APP switch is enabled and CVS-1450 device is reset.

### 2.6.2.3 Connectors

Peripherals	External Connectors	Function
Power	4 -Position power connector	Main power and power for isolated outputs.
IEEE-1394	6-pin IEEE 1394	Power and data connection to IEEE-1394 cameras
VGA	15-pin female DSUB(Standard VGA)	Video output
Serial	9-pin male DSUB(Standard RS-232 serial port)	COM1
10/100 Ethernet	RJ-45 (Standard Ethernet port)	Ethernet network connection
TRIG0	SMB receptacle	External isolated trigger input
TRIG1 & TRIG2	SMB receptacle	External TTL output
Digital Input/Output	44-pin female High density DSUB	External TTL I/O; External Isolated I/O

Table 2.3 CVS-1450 device connectors

# Chapter 3

## Configuration & application software development

---

### 3.1 Introduction to Vision Assistant 7.1

Vision Assistant is a tool for prototyping and testing image processing applications. To prototype an image processing application, build custom algorithms with the Vision Assistant scripting feature. The scripting feature records every step of the processing algorithm. After completing the algorithm, we can test it on other images to make sure it works. The algorithm is recorded in a Builder file, which is an ASCII text file that lists the processing functions and relevant parameters for an algorithm that we prototype in Vision Assistant. Using the LabVIEW VI Creation Wizard, we can create a LabVIEW VI that performs the prototype that we created in Vision Assistant. We must have LabVIEW 6.1 or later and IMAQ Vision 7.1 for LabVIEW or later installed to use the LabVIEW VI Creation Wizard.

### 3.2 Problem definition

IC is one the most commonly used circuit components in any electronic device. In Electronic Industry at the time of assembly of IC packages in an electronic circuit, it is very difficult to count the pins manually and also at the time of inventory check the classification of ICs is must so that particular IC packages go to their particular storage places. At the time Quality Control Check in an Electronic circuit correct IC selection has very important consideration. During Assembly, Inventory and Quality check it may happen that a wrong IC selected or IC package may have broken pin or pins or pins may be fused together are some of the problems of Electronic Circuit Manufacturing industry.

In IC package classification we must take care of position of IC and background disturbance. The script should work in all types of conditions such as IC might be positioned in horizontal, vertical, or in inclined position in real time and Ground plane may contain many types of blurs that affects the correct count of pins. Algorithm should be of such type that it counts correct number if pins in all conditions independent of disturbances. The algorithm must be adaptable to all these real time difficulties

### **3.3 Problem solution and its algorithm**

We are taking real time continuous images of ICs and apply the following algorithm on these different ICs. The same algorithm will find out correct number of pins present in 8, 14, 16, 62, 40, and 80 pin ICs, independent of their orientation. It will count correct number of pins in horizontal, vertical or in inclined position. Say if one or two pins are broken or fused together, will change the pin count of an IC package. So with the help of Machine Vision System we can count number of pins in an IC package and can identify above faults in IC package. Before acquiring images from camera there are some CVS and Camera adjustments to be done. Steps involving camera and CVS settings are given below [10].

#### **3.3.1 Configuring the system**

Before acquiring images we have to configure CVS-1450 and Camera. Steps involving in configuration of CVS and camera are:

##### **3.3.1.1 Configuring CVS-1450**

- Open Measurement and Automation Controller.
- On left side window click Remote systems.
- Select CVS 1450.
- Set the configuration of CVS.
- Buttons appear on screen will be: Reboot, Lock, Unlock, Refresh.
- Lock and unlock is used for the safety purposes so that no one could change your required configuration. It takes password to lock and unlock.
- IP address of camera must be entered in the IP address option during CVS settings.
- After fixing the required IP address we must lock this configuration.
- After CVS settings now select Devices and Interfaces.

##### **3.3.1.2 Configuring prosilica camera 2.0.1**

- Select camera (Cam 0 Prosilica 2.0.1).
- To configure camera select Camera Attributes.
- In camera Attributes select video mode as Format 7.
- Output Format: We can use cameras with the following output formats:
  - Monochrome (8 bits/pixel)
  - Monochrome (16 bits/pixel)
  - RGB (24 bits/pixel)

- RGB (48 bits/pixel)
  - YUV 4:1:1 (12 bits/pixel)
  - YUV 4:2:2 (16 bits/ pixel)
  - YUV 4:4:4 (24 bits/ pixel)
  - Bayer (8 bits/pixel)
  - Bayer (16 bits/pixel)
- In camera settings we can change Brightness, Contrast Color Plane etc in real time.
  - If the live image has any type of problem, it can be corrected by Camera settings.
  - Select Grab or Snap function as required.
  - It will open directly image acquire by camera in MAX.
  - If we are getting live image in Max, only then it will be taken by Vision Assistant 7.1.

### **3.3.2 Image acquisition**

- Open VA 7.1.
- In select network target write IP address of the computer you want to use to acquire images.
- Select Video mode Format 7.
- We have four Buttons here: Acquire single image, Acquire continuous image, Refresh list of devices, and Store acquired image in browser.
- Select Refresh list of devices button.
- Camera name will display in window below this button.
- Select Cam0, the name of the camera.
- Image will appear in the browser window and using other three images we can snap or grab the picture in real time.
- For improving image clarity camera is provided with two settings:
  1. Focus setting
  2. Aperture Setting
- Camera height is one of the most important factors to be considered in mind. Camera height should be adjusted such that it takes the entire view of subject.
- After this we can apply image-processing tools in real time and get the required information from image.

### 3.3.3 Process image

Image processing is done using compact vision system 1450. Steps involving processing are given below:

- **Acquire image**

We acquire Image of specimen by IEEE-1394 Prosilica 2.0.1 Camera.

- **Extract color plane**

We first extract any one color plane from the image so that image can become 8-bit. We extract red plane from the image.

- **Image masking**

It will mask the non desired part of the image. We select region of interest (ROI) in the image and then do image processing functions.

- **Threshold**

It will help in extracting selected image portion of interest. In this case we extract pins by applying required threshold and it will show selected region in Red color.

- **Morphological operations**

We use different operations from Binary and advance morphological functions to get the desired output.

- **Particle analysis**

It counts the number of cells present in processed image.

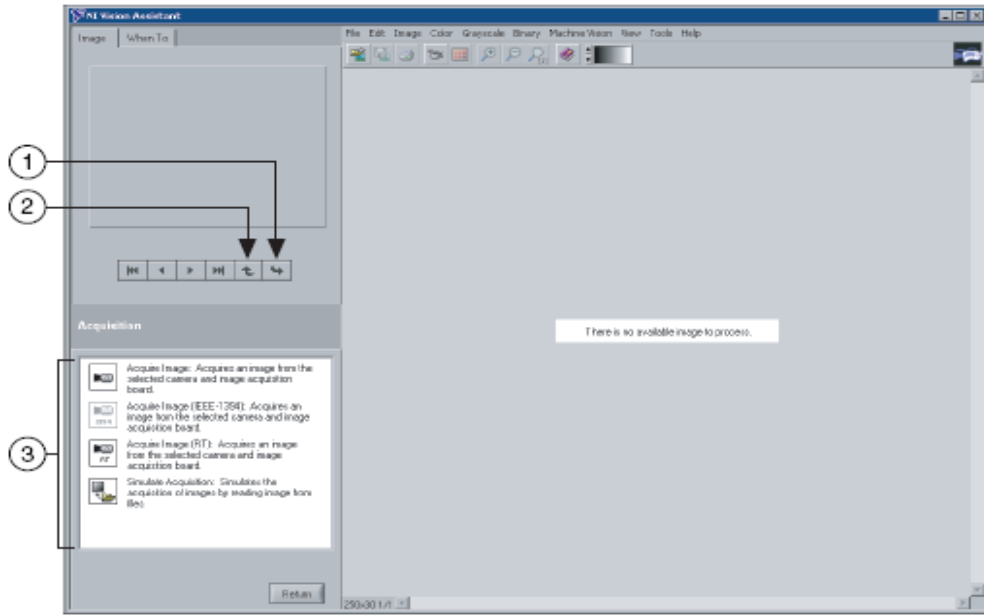
## 3.4 Software development in Vision Assistant 7.1

Vision Assistant loads the image files into the Image Browser. The Image browser provides information about the selected image, such as image size, location, and type. Following are the steps to perform image processing an analysis and corresponding windows are shown.

### 3.4.1 Pin counting (40 pin IC) in vertical direction

#### 3.4.1.1 Welcome screen

Image Browser contains all of the images currently loaded in Vision Assistant. We can select an image to process by double-clicking it in the Image Browser and processing window updates the image as we change parameters. Because this view immediately reflects the changes we have made in the Parameter window, we can continue modifying parameters until we get the result we want.



1. Make image active
2. Store acquired image in browser button
3. Acquisition functions

Fig. 3.1 Acquiring images in Vision Assistant

### 3.4.1.2 Image acquisition

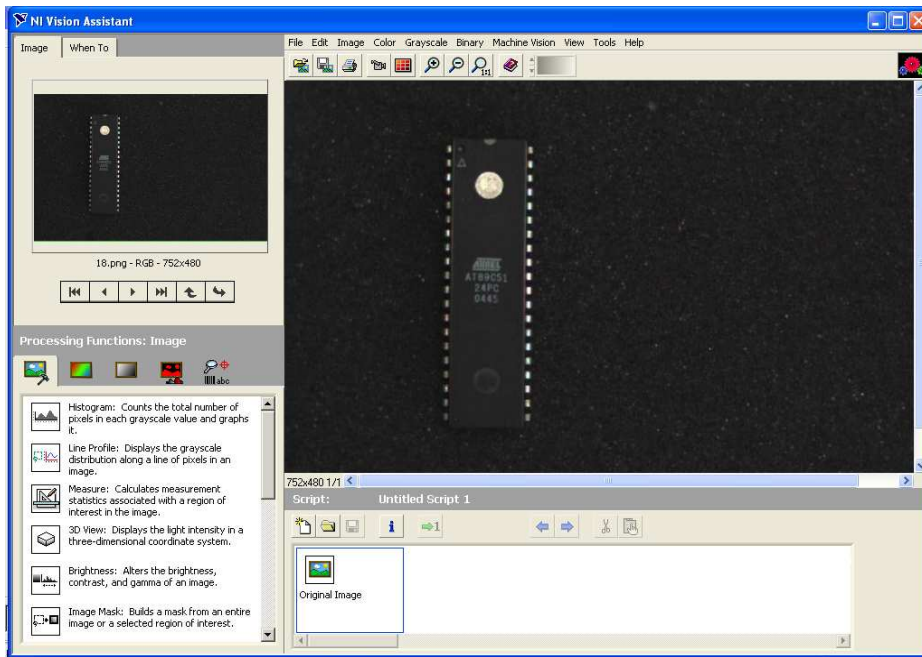


Figure 3.2 Opening an image

### 3.4.1.3 Extract color plane

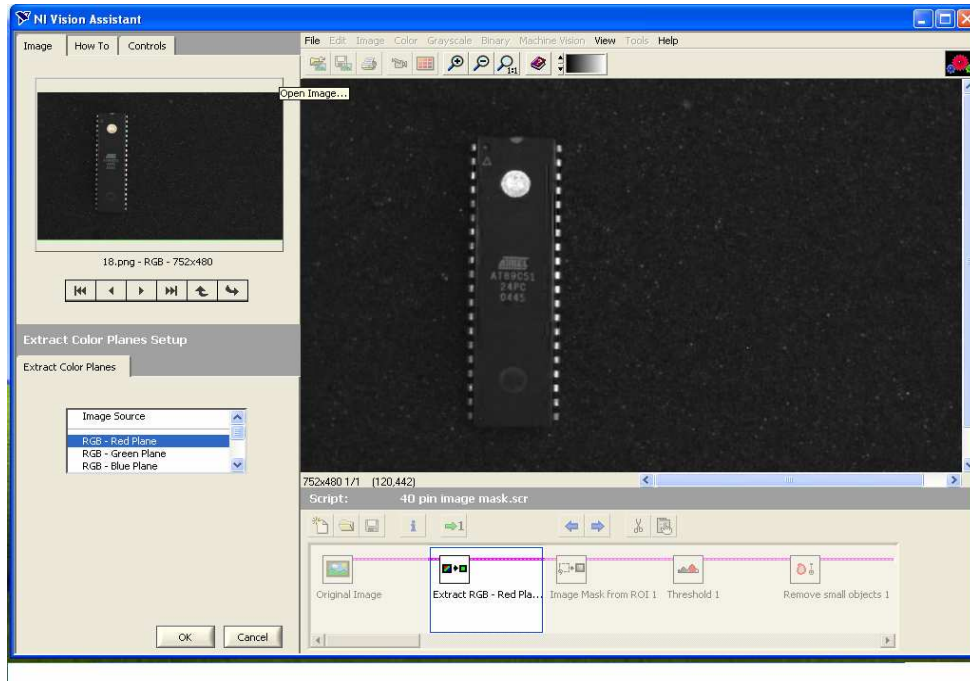
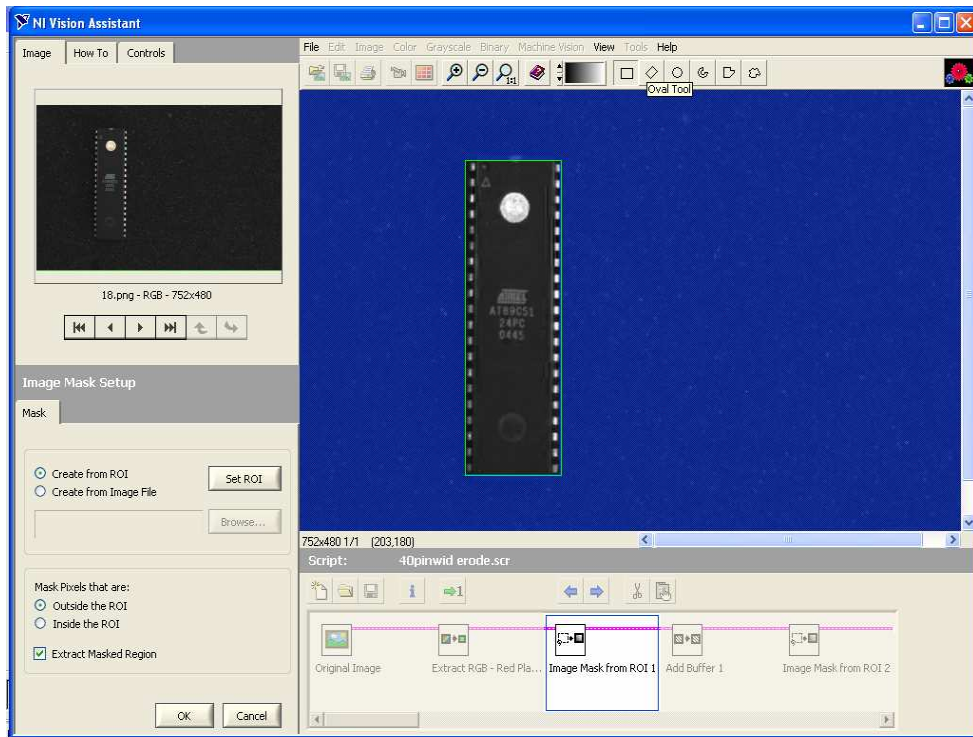


Figure 3.3 Extracting color plane

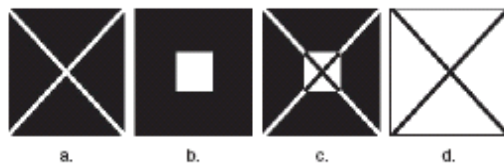
### 3.4.1.4 Image masking

An image mask isolates parts of an image for processing. If a function has an image mask parameter, the function process or analysis depends on both the source image and the image mask. An image mask is an 8-bit binary image that is the same size as or smaller than the inspection image. Pixels in the image mask determine whether corresponding pixels in the inspection image are processed. If a pixel in the image mask has a nonzero value, the corresponding pixel in the inspection image is processed. If a pixel in the image mask has a value of 0, the corresponding pixel in the inspection image is not processed. We use image masks when we want to focus our processing or inspection on particular regions in the image.



*Figure 3.4 Image masking*

Pixels in the source image are processed if corresponding pixels in the image mask have values other than zero. Figure 3.5 shows how a mask affects the output of the function that inverts the pixel values in an image. Figure 3.5a shows the inspection image. Figure 3.5b shows the image mask. Pixels in the mask with zero values are represented in black, and pixels with nonzero values are represented in white. Figure 3.5c shows the inverse of the inspection image using the image mask. Figure 3.5d shows the inverse of the inspection image without the image mask.



*Figure 3.5 The effect of an image mask*

We can limit the area in which our function applies an image mask to the bounding rectangle of the region we want to process. This technique saves memory by limiting the image mask to only the part of the image containing significant information. To keep track of the location of this region of interest (ROI) in regard to the original image, IMAQ Vision sets an offset. An offset defines the coordinate position in the

original image where we want to place the origin of the image mask. Figure 3.6 illustrates the different methods of applying image masks. Figure 3.6a shows the ROI in which we want to apply an image mask. Figure 3.6b shows an image mask with the same size as the inspection image. In this case, the offset is set to  $[0, 0]$ . A mask image also can be the size of the bounding rectangle of the ROI, as shown in Figure 3.6c, where the offset specifies the location of the mask image in the reference image. We can define this offset to apply the mask image to different regions in the inspection image.

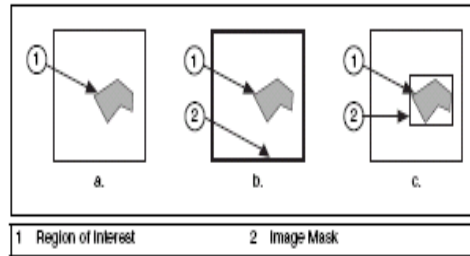


Figure 3.6 Selecting ROI

Figure 3.7 illustrates the use of a mask with two different offsets. Figure 3.7a shows the inspection image, and Figure 3.7b shows the image mask. Figure 3.7c and Figure 3.7d show the results of a function using the image mask given the offsets of  $[0, 0]$  and  $[3, 1]$ , respectively.

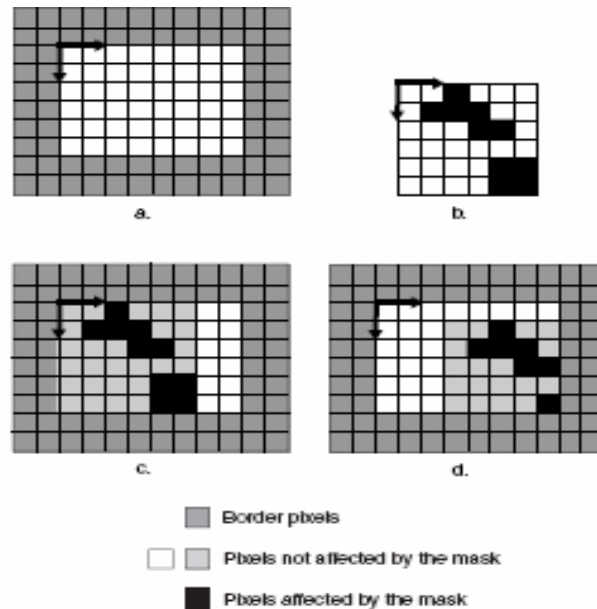


Figure 3.7 Effect of applying a mask with different offsets

### 3.4.1.5 Store image

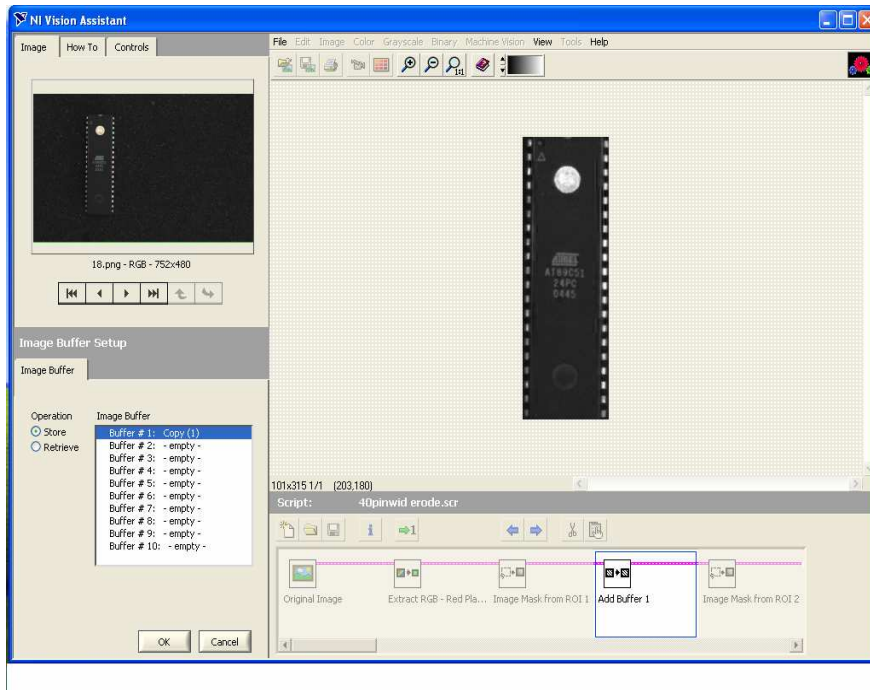


Figure 3.8 Storing an image in buffer

### 3.4.1.6 Image masking

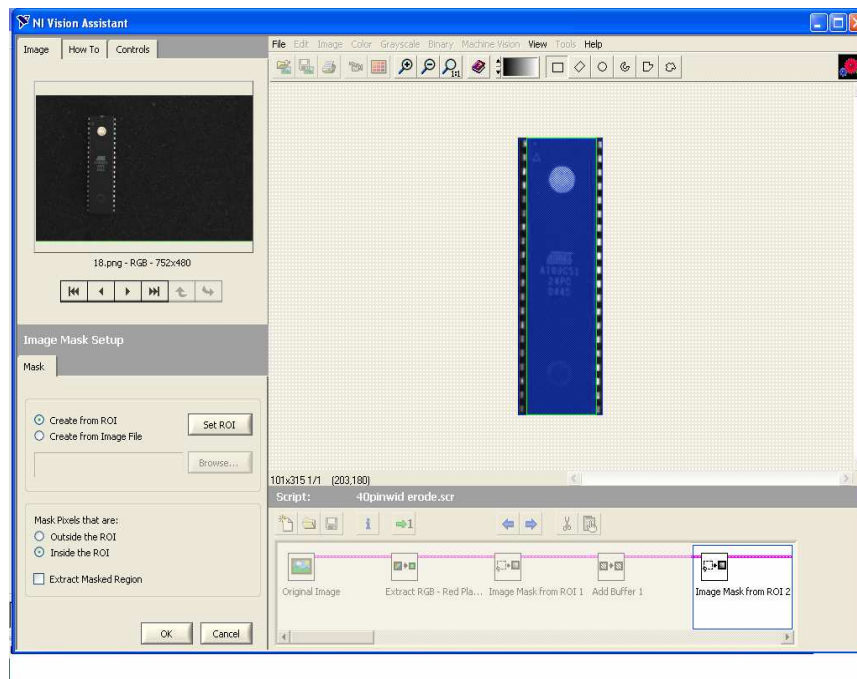
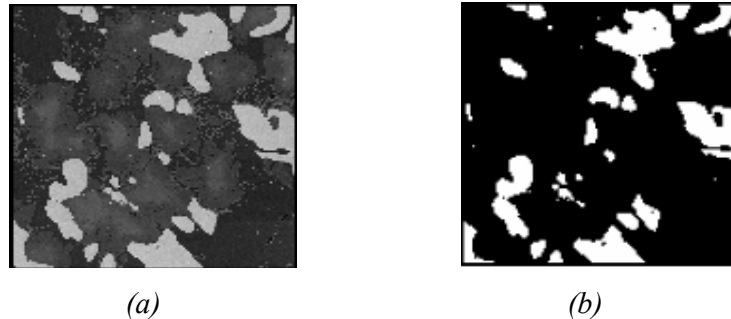


Figure 3.9 Image masking

### 3.4.1.7 Threshold

Color threshold converts a color image into a binary image. A color threshold is applied when we have to isolate features for analysis and processing, or to remove unnecessary features. Before performing a color threshold, we may need to enhance our image with lookup tables or the equalize function. To threshold a color image, specify a threshold interval for each of the three-color components.



*Figure 3.10 Effect of threshold*

A pixel in the output image is set to 1 if and only if its color components fall within the specified ranges. Otherwise, the pixel value is set to 0. In figure 3.10(a), the gray shaded region indicates the threshold range for each of the color planes. For a pixel in the color image to be set to 1 in the binary image, its red value should lie between 130 and 200, its green value should lie between 100 and 150, and its blue value should lie between 55 and 115.

**Threshold Ranges for an HSL Image** The hue plane contains the main color information in an image. To threshold an HSL image, first determine the hue values of the pixels that we want to analyze after threshold. In some applications, we may need to select colors with the same hue value but various saturation values. Because the luminance plane contains only information about the intensity levels in the image, we can set the luminance threshold range to include all the luminance values, thus making the threshold process independent from intensity information.

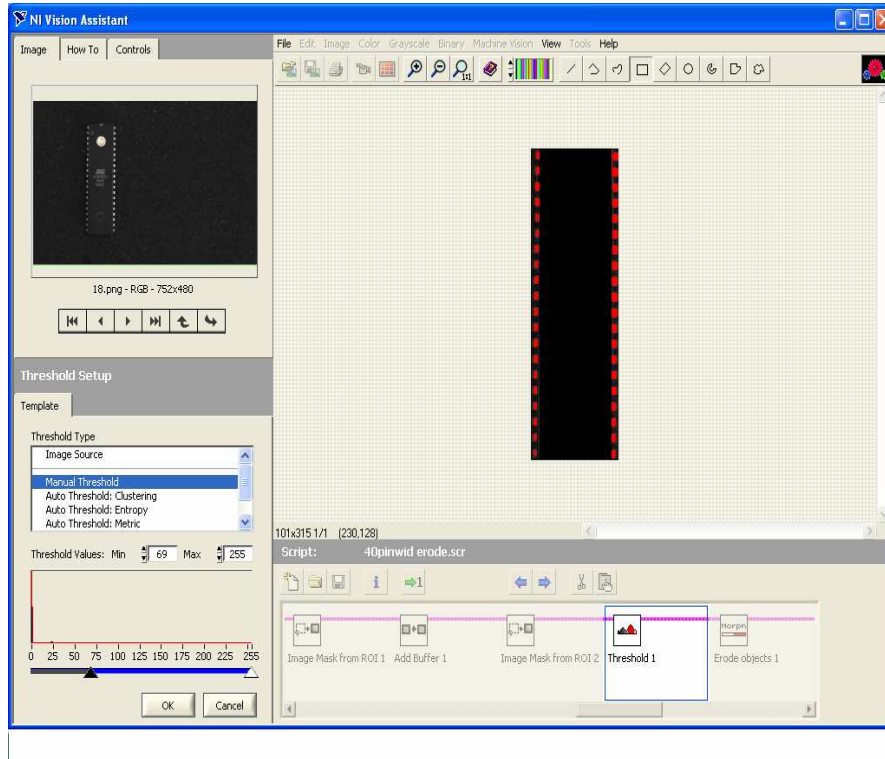


Figure 3.11 Applying threshold

### 3.4.1.8 Erosion

Erosion is typically applied to binary images. The basic effect of the operator on a binary image is to erode away the boundaries of regions of foreground pixels (i.e. white pixels, typically). Thus areas of foreground pixels shrink in size, and holes within those areas become larger. The original image is shown in Fig.3.12 (a) and in Fig.3.12 (b) image is the result of eroding four times with a disk shaped structuring element which is 11 pixels in diameter is shown in Fig.3.12 (b).



(a) Original image



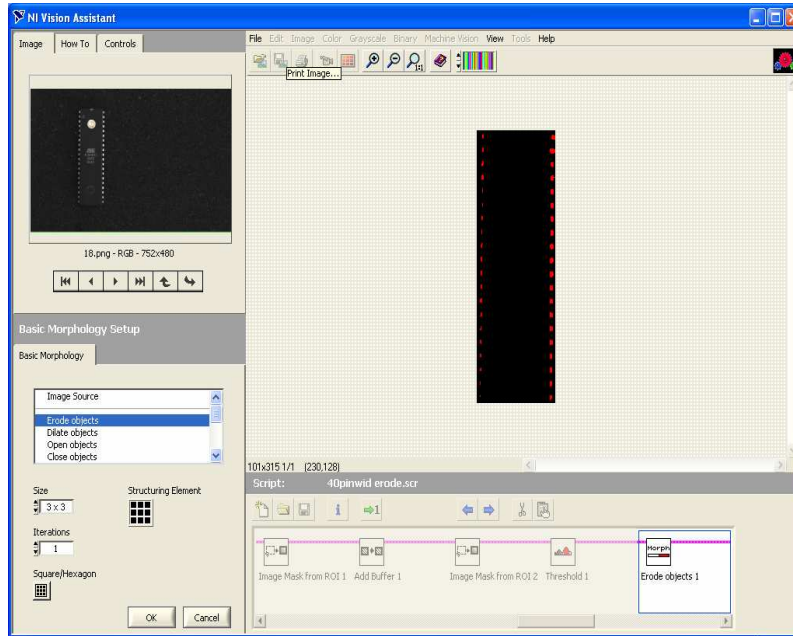
(b) Eroded image

Figure 3.12 Effect of erosion

It shows that the hole in the middle of the image increases in size as the border shrinks. Note that the shape of the region has been quite well preserved due to the use of a disk shaped structuring element.

In general, erosion using a disk shaped structuring element will tend to round concave boundaries, but will preserve the shape of convex boundaries. Since erosion removes one pixel from around the object, the object becomes increasingly thinner with each pass. It will remove one pixel from the perimeter (and holes) of the object even if the shape of the object is eventually lost. The final result of too many types of erosion will be the loss of the object. That is to say, if the reversing operation of dilation, which adds one pixel to the perimeter of the object with each pass, is used, the dilated may not resemble the original object at all. In fact, if the object is totally eroded to one pixel, dilation will result in a square or a circle. As a result, erosion can irreparably damage the image. However, it can also be successfully used to eliminate unwanted objects in an image. For example, if identification of the largest object in an image is needed, than successive erosions will eliminate all other objects before the largest is eliminated. Thus the object of interest can be identified.

The gray-level morphology functions apply to gray-level images. We can use these functions to alter the shape of regions by expanding bright areas at the expense of dark areas and vice versa. These functions smooth gradually varying patterns and increase the contrast in boundary areas. This section describes the following gray-level morphology functions. These functions are derived from the combination of gray-level erosions and dilations that use structuring element .A gray-level erosion reduces the brightness of pixels that are surrounded by neighbors with a lower intensity. The neighborhood is defined by a structuring element.



### 3.13 Erosion

#### 3.4.1.9 Dilation

Dilation is one of the two basic operators in the area of mathematical morphology the other being erosion. It is typically applied to binary images. The basic effect of the operator on a binary image is to gradually enlarge the boundaries of regions of foreground pixels (i.e. white pixels, typically). Thus areas of foreground pixels grow in size while holes within those regions become smaller. The original image is shown in Fig.3.14 (a). The basic effect of dilation is shown in Fig.3.14 (b).



(a) Original image

(b) Dilated image

Figure 3.14 Effect of dilation

Two dilation passes using a disk shaped structuring element of 11 pixels radius produce this image. The corners get rounded off. In general, when dilating by a disk shaped structuring element, convex boundaries will become rounded, and concave

boundaries will be preserved as they are. With additional applications of dilation, the objects, as well as the disappearing hole, can become one solid piece and hence cannot be recognized as distinct objects. Gray-level dilation increases the brightness of each pixel that is surrounded by neighbors with a higher intensity. The neighborhood is defined by a structuring element. The gray-level dilation has the opposite effect of the gray-level erosion because dilating bright regions also erodes dark regions.

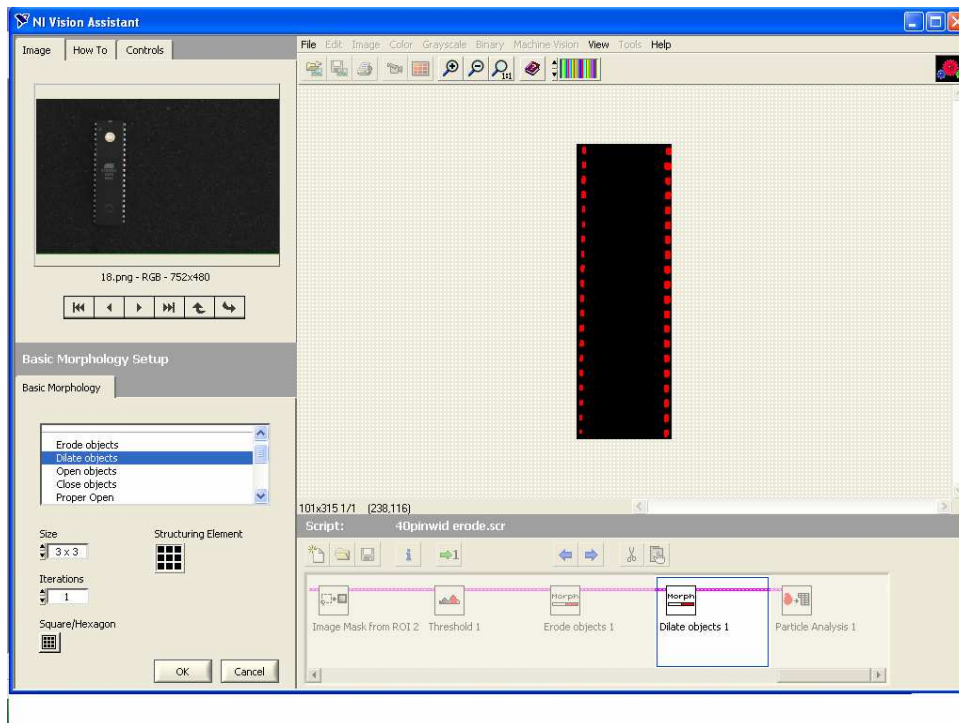


Figure 3.15 Dilation

### 3.4.1.10 Particle filter

A particle filter isolates and keeps the circular particles and removes the non-circular particles from the image. We are using Heywood Circularity Factor from the list of particle filters. This function calculates the ratio of the perimeter of the particle to the perimeter of the circle with the same area. The more circular the particle, the more close the ratio to 1.

Measurement	Definition	Equation
Heywood Circulatory Factor	Perimeter divided by the circumference of the circle with the same area. The closer the shape of a particle is to a disk, the closer the Heywood Circulatory Factor to 1.	$P / 2\sqrt{\pi A}$

Table 3.1 Factor

In the equation given above A is the area of particle and P is the perimeter (Length of the outer boundary of the particle. Because the boundary is comprised of discrete pixels, Software sub samples the boundary points to approximate a smoother, more accurate perimeter.

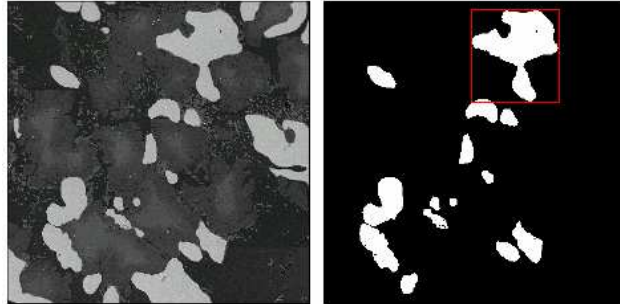
#### **3.4.1.11 Particle analysis**

We can use particle analysis to detect connected regions or groupings of pixels in an image and then make selected measurements of those regions. These regions are commonly referred to as particles. A particle is a contiguous region of nonzero pixels. We can extract particles from a grayscale image by threshold the image into background and foreground states. Zero valued pixels are in the background state, and all nonzero valued pixels are in the foreground. Particle analysis consists of a series of processing operations and analysis functions that produce information about particles in an image. Using particle analysis, we can detect and analyze any 2D shape in an image.

To find the area occupied by these circles we use particle analysis. Particle analysis is used when we are interested in finding particles whose spatial characteristics satisfy certain criteria. In many applications where computation is time-consuming, we can use particle filtering to eliminate particles that are of no interest based on their spatial characteristics, and keep only the relevant particles for further analysis. We can use particle analysis to find statistical information such as the presence of particles, their number and size, and location. This information allows us to perform many machine vision inspection tasks such as detecting flaws on silicon wafers, detecting soldering defects on electronic boards, or web inspection applications such as finding structural defects on wood planks or detecting cracks on plastics sheets. We also can locate objects in motion control applications.

In applications where there is a significant variance in the shape or orientation of an object, particle analysis is a powerful and flexible way to search for the object. We can use a combination of the measurements obtained through particle analysis to define a feature set that uniquely defines the shape of the object. A typical particle analysis process scans through an entire image, detects all the particles in the image, and builds a detailed report on each particle. We can use multiple parameters such as perimeter, angle, area, and center of mass to identify and classify these particles. Using multiple parameters can be faster and more effective than pattern matching in many applications. In addition, by using different sets of parameters, we can uniquely identify a feature in an image. For

example, we could use the area of the template particle as a criterion for removing all particles that do not match it within some tolerance. We then can perform a more refined search on the remaining particles using another list of parameter tolerances. The following figure shows a sample list of parameters that we can obtain in a particle analysis application. The binary image in this example was obtained by threshold the source image and removing particles that touch the border of the image.



*Fig. 3.16 Selection of area of an image to analyze*

We can use these parameters to identify and classify particles. The following table 3.3 shows the values obtained for the particle enclosed in a rectangle, shown in the Fig. 3.16.

Particle Measurements	Values
Area	2456
Number of holes	1
Bounding Rectangle	
Left	127
Top	8
Right	200
Bottom	86
Center of mass	
X	167.51
Y	37.61
Orientation	82.36°
Dimension	
Width	73
Height	78

*Table 3.2 Particle measurements*

To use particle analysis, first create a binary image using a thresholding process. We then can improve the binary image using morphological transformations and make measurements on the particles in the image.

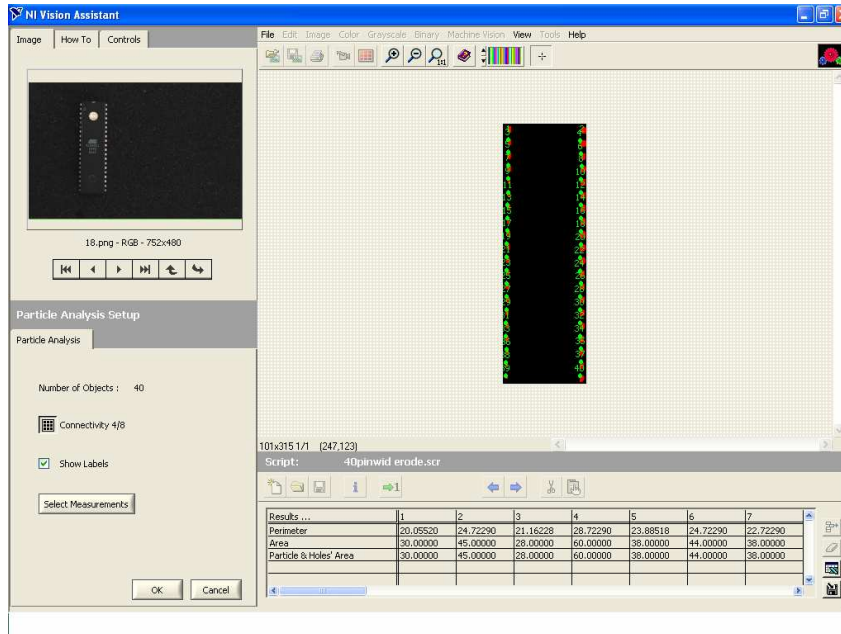


Figure 3.17 Particle analysis

### 3.4.1.12 Result

Object	Normalized Perimeter	Normalized Area	Normalized Particle & Holes' Area	Number of Holes
1	20.0552	30	30	0
2	24.7229	45	45	0
3	21.16228	28	28	0
4	28.7229	60	60	0
5	23.88518	38	38	0
6	24.7229	44	44	0
7	22.7229	38	38	0
8	25.55133	50	50	0
9	17.9907	21	21	0
10	26.32456	48	48	0
11	16.0645	18	18	0
12	26.32456	48	48	0
13	19.48683	27	27	0
14	26.7229	51	51	0
15	17.9907	21	21	0
16	26.7229	52	52	0
17	21.55133	32	32	0
18	27.98141	56	56	0

19	20.7229	31	31	0
20	28.38905	57	57	0
21	22.32456	31	31	0
22	28.0552	58	58	0
23	21.16228	28	28	0
24	26.32456	48	48	0
25	21.15298	34	34	0
26	24.7229	45	45	0
27	21.15298	30	30	0
28	25.15298	46	46	0
29	20.32456	25	25	0
30	26.32456	46	46	0
31	20.32456	27	27	0
32	26.32456	46	46	0
33	17.56062	24	24	0
34	23.65685	40	40	0
35	25.56062	43	43	0
36	17.9907	23	23	0
37	23.88518	42	42	0
38	16.0645	18	18	0
39	14	12	12	0
40	23.65685	37	37	0

*Table 3.3 Pin count of 40 pin IC (Vertical)*

#### **3.4.1.13 Estimating processing time**

Vision Assistant can estimate the time, in milliseconds, that Software will take to process the active image with the open script. The Performance Meter gives both an estimate of the total time Software will take to process the image and an estimate of the time each function within the script will require. Below shown is performance meter which indicates time taken at each step by VA 7.1.

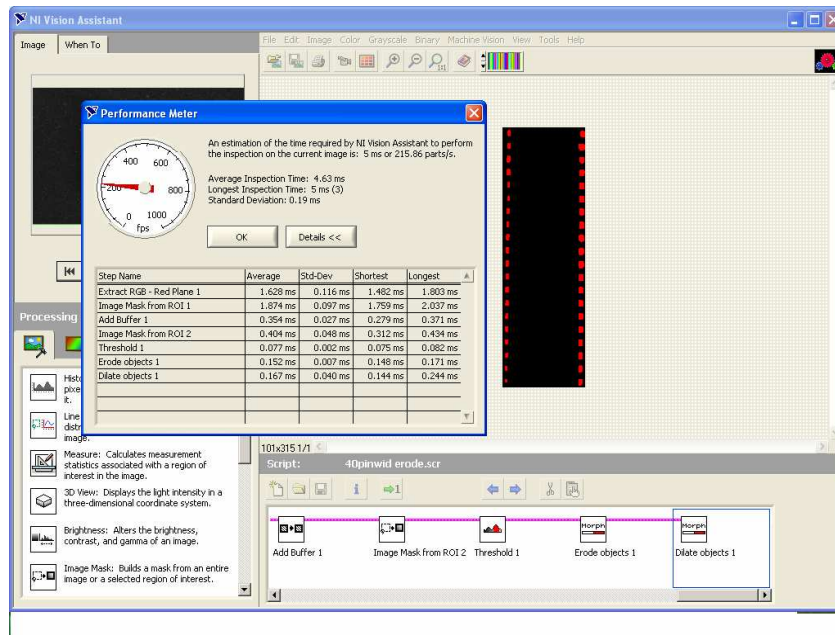


Figure 3.18 Estimated times

### 3.4.1.14 Performance meter

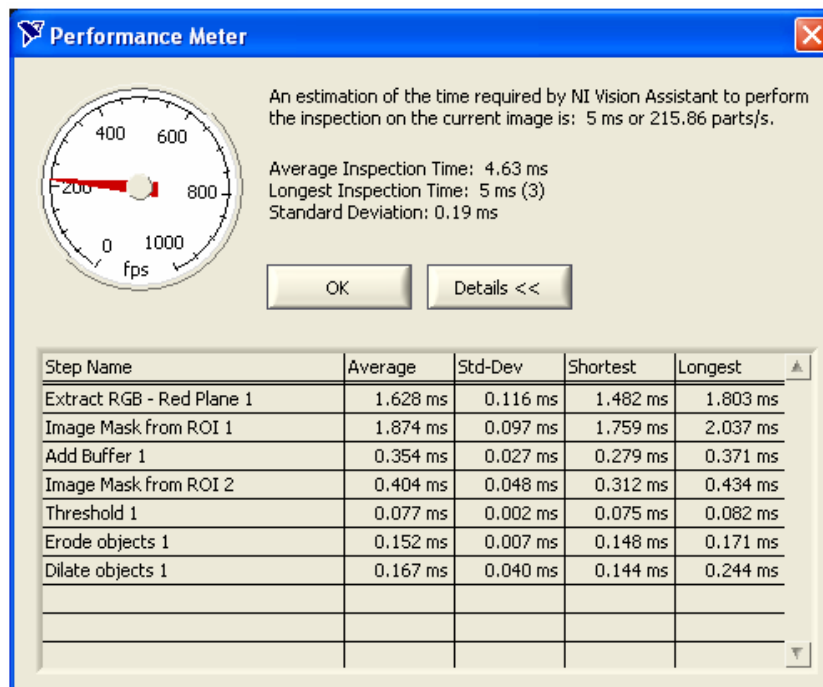


Figure 3.19 Performance meter

### 3.4.2 Pin counting (40 pin IC) in horizontal direction

#### 3.4.2.1 Acquire image

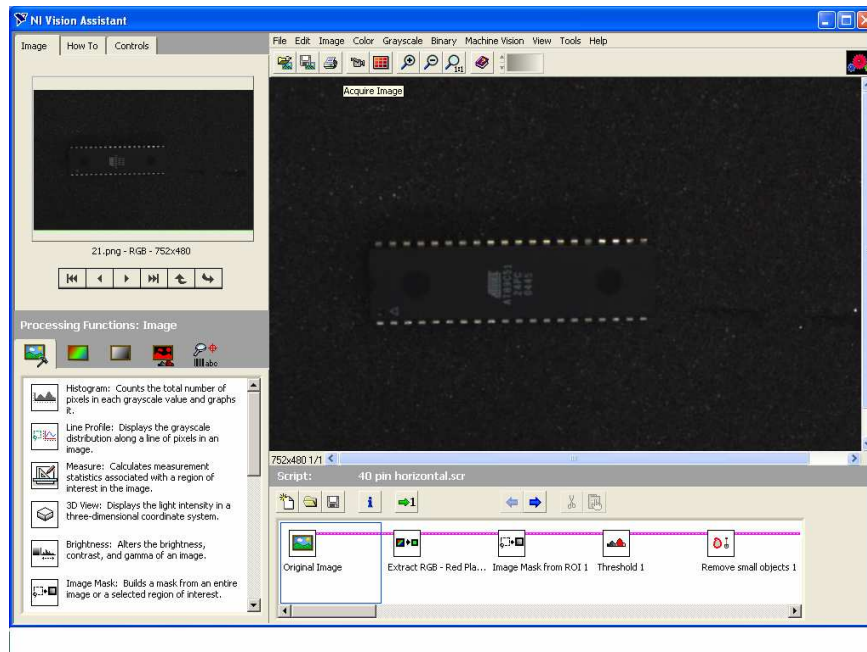


Figure 3.20 Image acquisition

#### 3.4.2.2 Particle analysis

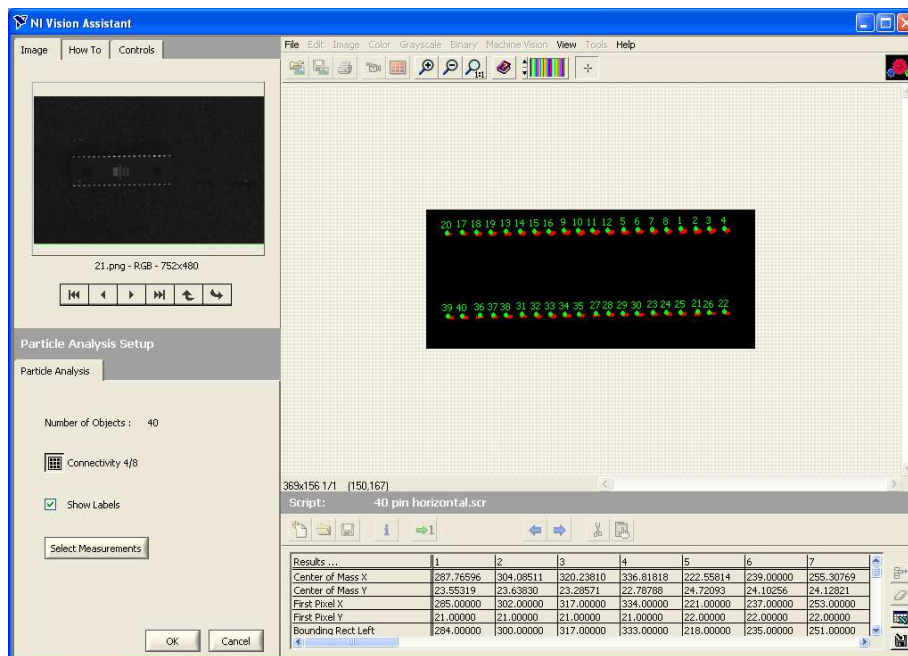


Figure 3.21 Particle analysis

### 3.4.2.3 Results

Object	Normalized Perimeter	Normalized Area	Normalized Particle & Holes' Area	Number of Holes
1	25.47754	47	47	0
2	24.37976	47	47	0
3	24.38905	42	42	0
4	22.32456	33	33	0
5	24.71361	43	43	0
6	23.98141	39	39	0
7	23.47754	39	39	0
8	20.80984	34	34	0
9	24.37976	44	44	0
10	24.37976	46	46	0
11	23.55133	42	42	0
12	23.47754	40	40	0
13	23.04746	33	33	0
14	21.88518	36	36	0
15	23.88518	37	37	0
16	23.22677	37	37	0
17	18.7229	25	25	0
18	22.7874	32	32	0
19	23.56062	34	34	0
20	16.32456	19	19	0
21	21.55133	30	30	0
22	20.0552	27	27	0
23	20.37976	30	30	0
24	20.0552	29	29	0
25	22	28	28	0
26	19.16228	22	22	0
27	19.94967	27	27	0
28	20.0552	28	28	0
29	20.32456	27	27	0
30	19.55133	29	29	0
31	21.62512	27	27	0
32	22.88363	35	35	0
33	21.47754	34	34	0
34	22.7229	33	33	0
35	19.55133	27	27	0
36	22.89292	28	28	0
37	19.21748	27	27	0
38	20.7229	30	30	0
39	19.80326	29	29	0
40	22.7229	31	31	0

Table 3.4 Pin count of 40 pin IC (Horizontal)

### 3.4.3 Pin counting (62 pin) in horizontal direction

#### 3.4.3.1 Acquire image

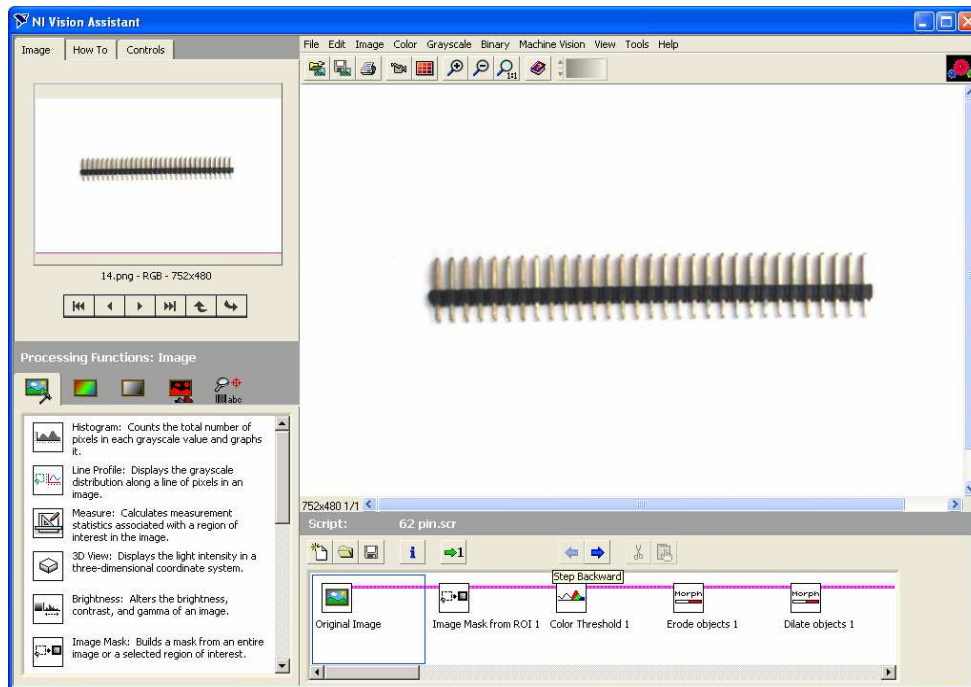


Figure 3.22 Image acquisition

#### 3.4.3.2 Results

Object	Normalized Perimeter	Normalized Area	Normalized Particle & Holes' Area	Number of Holes
1	65.86659	179	179	0
2	65.47754	182	182	0
3	65.47754	178	178	0
4	62.63052	168	168	0
5	64.29667	193	193	0
6	64.63982	177	177	0
7	64.29667	183	183	0
8	64.19114	186	186	0
9	64.31526	172	172	0
10	64.53429	190	190	0
11	64.31526	180	180	0
12	62.71361	183	183	0
13	61.87588	147	147	0
14	60.71361	176	176	0
15	61.54203	184	184	0
16	59.02887	165	165	0
17	60.63052	159	159	0
18	61.54203	170	170	0

19	63.54203	171	171	0
20	60.36116	166	166	0
21	60.71361	176	176	0
22	57.20818	157	157	0
23	58.97367	166	166	0
24	57.20818	167	167	0
25	59.80209	162	162	0
26	60.95508	185	185	0
27	57.47754	141	141	0
28	57.47754	148	148	0
29	56.63982	154	154	0
30	52.71361	129	129	0
31	53.03816	123	123	0
32	25.47754	47	47	0
33	24.37976	47	47	0
34	24.38905	42	42	0
35	22.32456	33	33	0
36	24.71361	43	43	0
37	23.98141	39	39	0
38	23.47754	39	39	0
39	20.80984	34	34	0
40	24.37976	44	44	0
41	24.37976	46	46	0
42	23.55133	42	42	0
43	23.47754	40	40	0
44	23.04746	33	33	0
45	21.88518	36	36	0
46	23.88518	37	37	0
47	23.22677	37	37	0
48	18.7229	25	25	0
49	22.7874	32	32	0
50	23.56062	34	34	0
51	16.32456	19	19	0
52	21.55133	30	30	0
53	20.0552	27	27	0
54	20.37976	30	30	0
55	20.0552	29	29	0
56	22.04746	28	28	0
57	19.16228	22	22	0
58	19.94967	27	27	0
59	20.0552	28	28	0
59	20.32456	27	27	0
60	19.55133	29	29	0
61	21.62512	27	27	0

*Table 3.5 Pin count of 62 pin device (Horizontal)*

### 3.4.4 Pin counting (8 pin IC in vertical direction)

#### 3.4.4.1 Acquire image

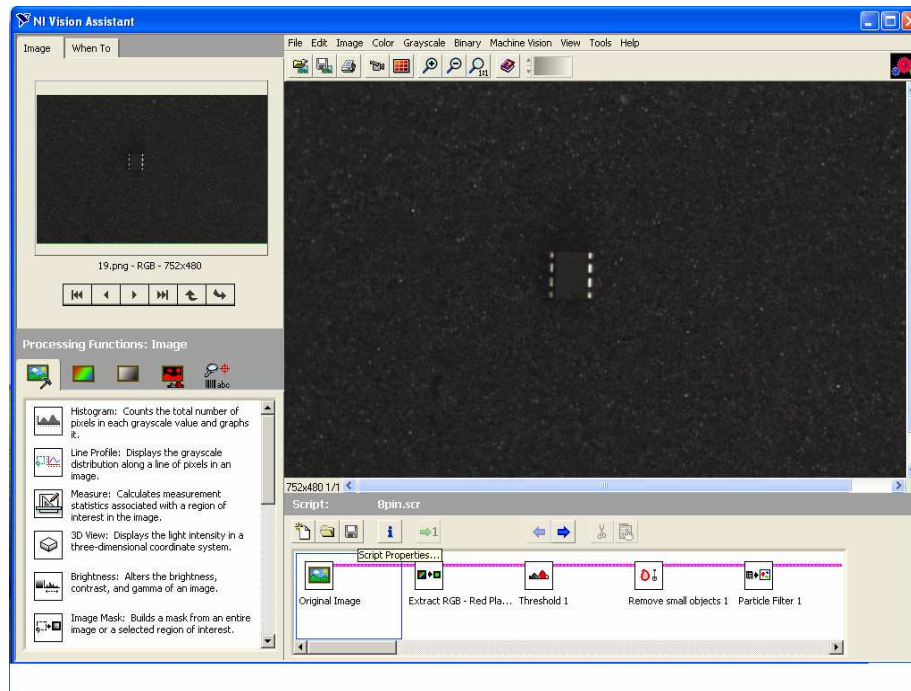


Figure 3.23 Image acquisition

#### 3.4.4.2 Particle analysis

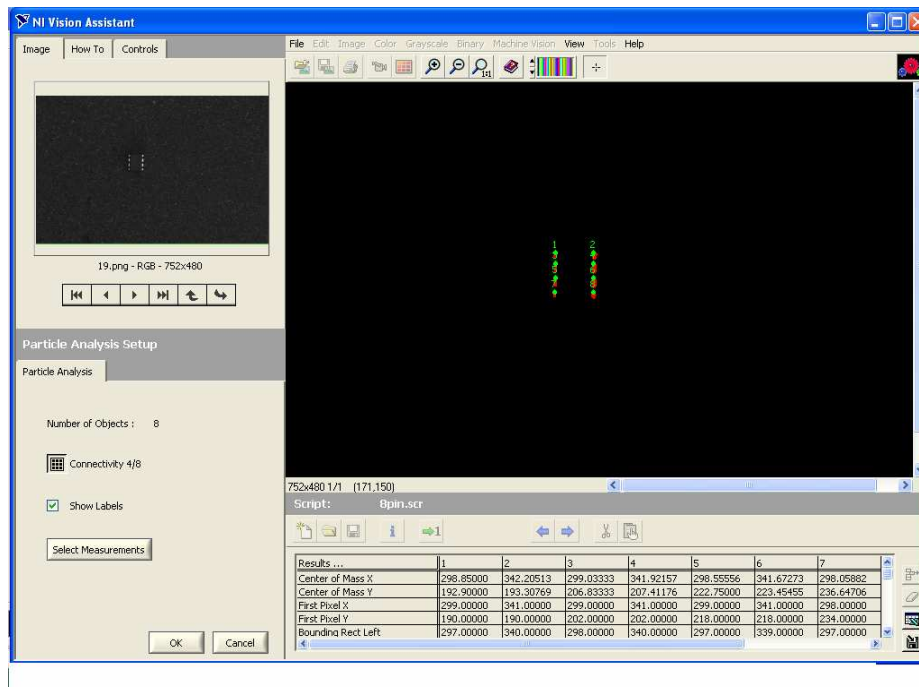


Figure 3.24 Particle analysis

### 3.4.4.3 Results

Object	Normalized Perimeter	Normalized Area	Normalized Particle & Holes' Area	Normalized Number of Holes
1	16.46284	20	20	0
2	22.7229	39	39	0
3	22.64911	30	30	0
4	28.45355	51	51	0
5	23.55133	36	36	0
6	29.20818	55	55	0
7	15.56062	17	17	0
8	20.88363	32	32	0

Table 3.6 Pin count of 8 pin IC (Vertical)

### 3.4.5 Pin counting (16 pin IC in inclined Position)

#### 3.4.5.1 Acquire image

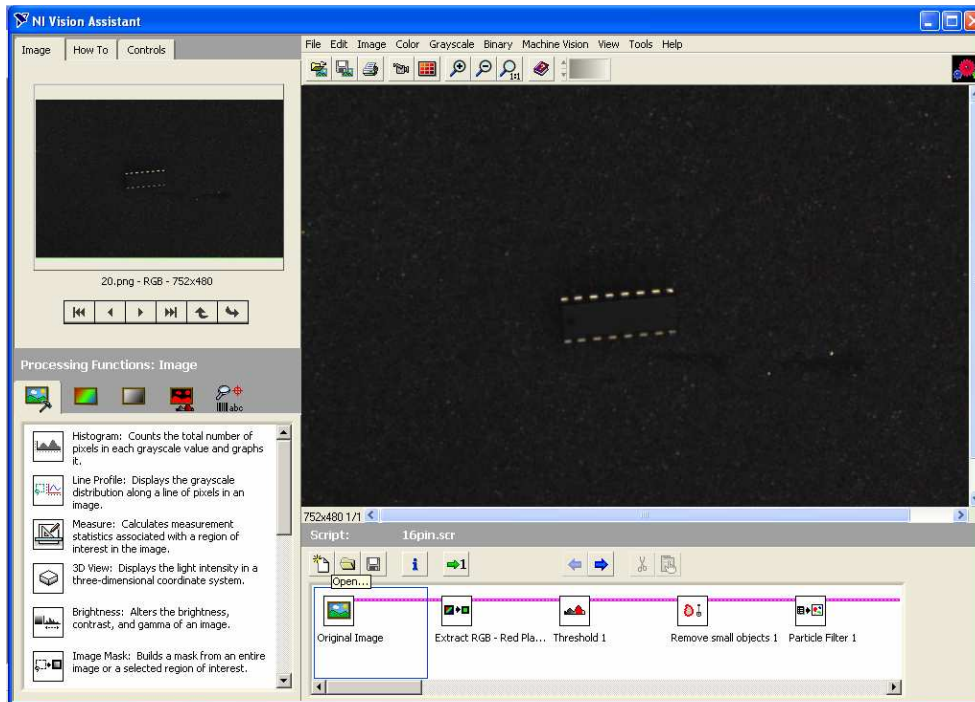


Figure 3.25 Image acquisition

### 3.4.5.2 Particle analysis

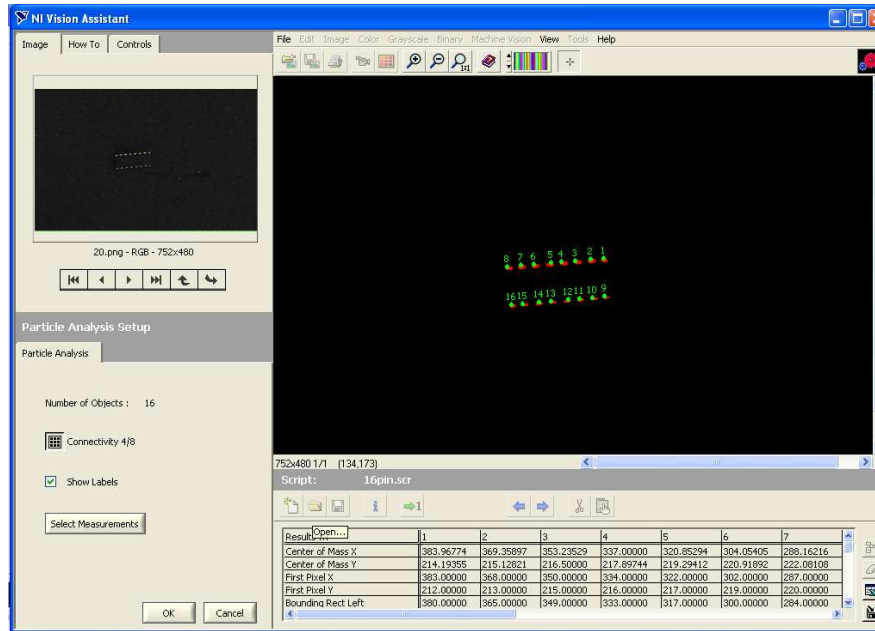


Figure 3.26 Particle analysis

### 3.4.5.3 Results

Object	Normalized Perimeter	Normalized Area	Normalized Particle & Holes' Area	Number of Holes
1	20.0552	31	31	0
2	22.0552	39	39	0
3	22.7229	34	34	0
4	23.47754	39	39	0
5	21.21748	34	34	0
6	22.80984	37	37	0
7	22.0552	37	37	0
8	20.0552	29	29	0
9	17.88518	18	18	0
10	20.32456	22	22	0
11	20.38905	25	25	0
12	18.88363	23	23	0
13	20.0552	23	23	0
14	20.0552	27	27	0
15	19.55133	25	25	0
16	20.38905	25	25	0

Table 3.7 Pin count of 16 pin IC (Inclined)

### 3.4.6 Pin counting (IC base in inclined position)

#### 3.4.6.1 Acquire image

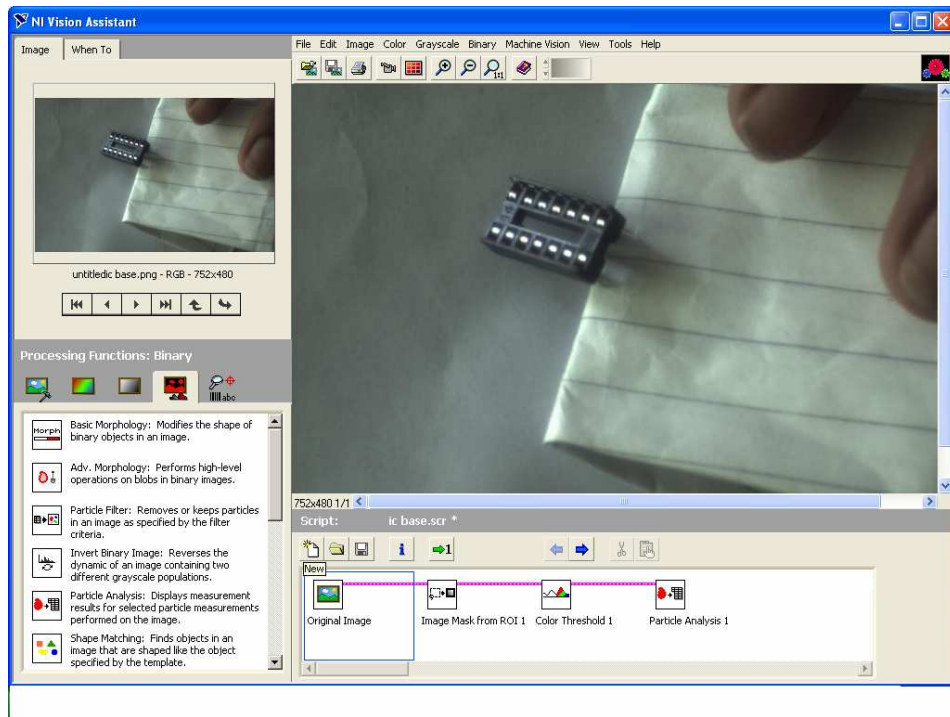


Figure 3.27 Image acquisition

#### 3.4.6.2 Image masking

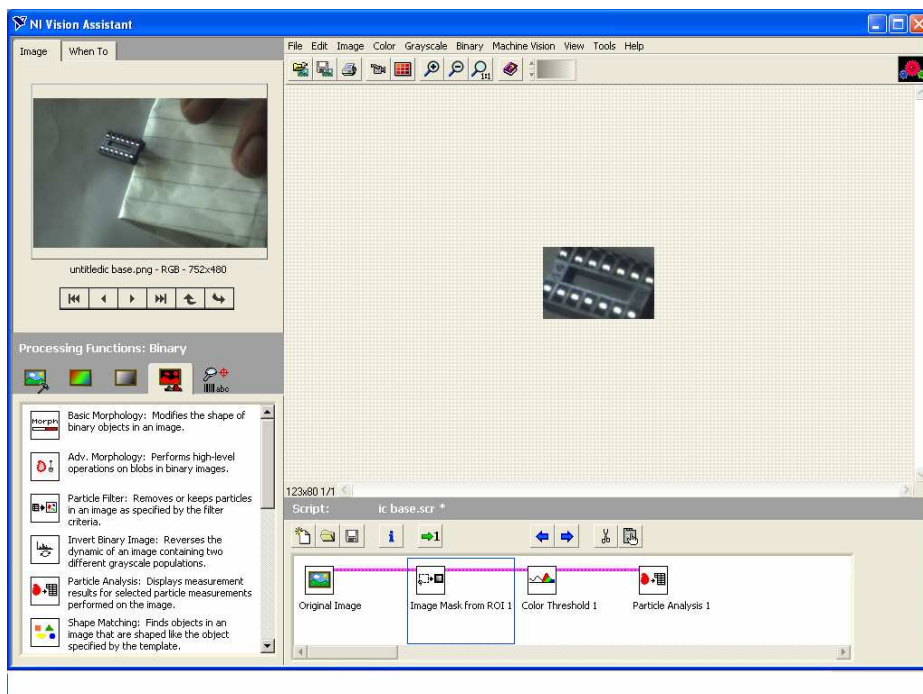


Figure 3.28 Image masking

### 3.4.6.3 Threshold

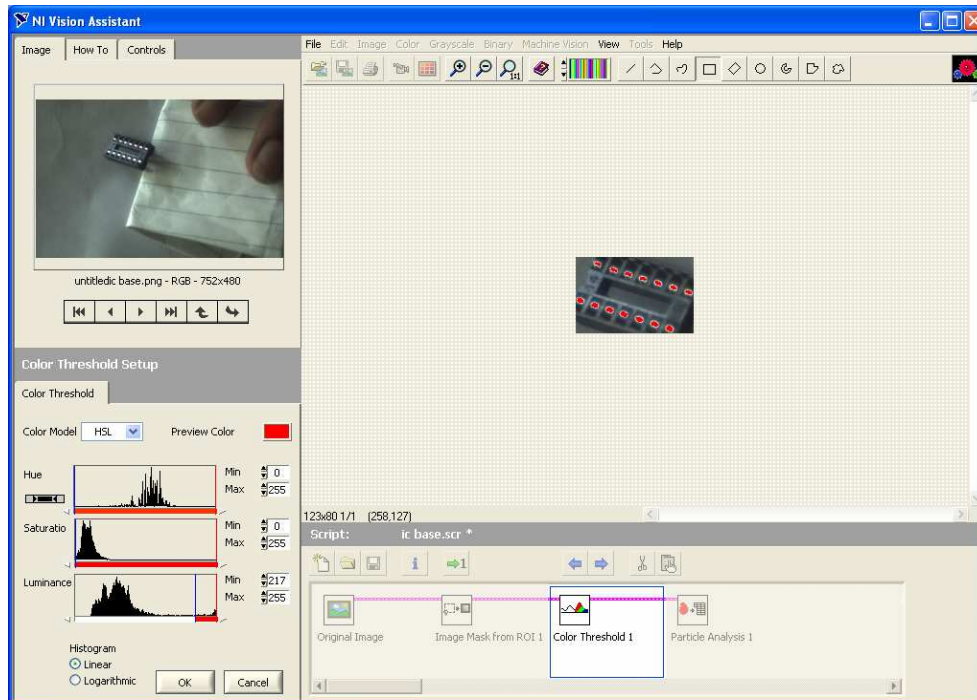


Figure 3.29 Threshold

### 3.4.6.4 Particle analysis

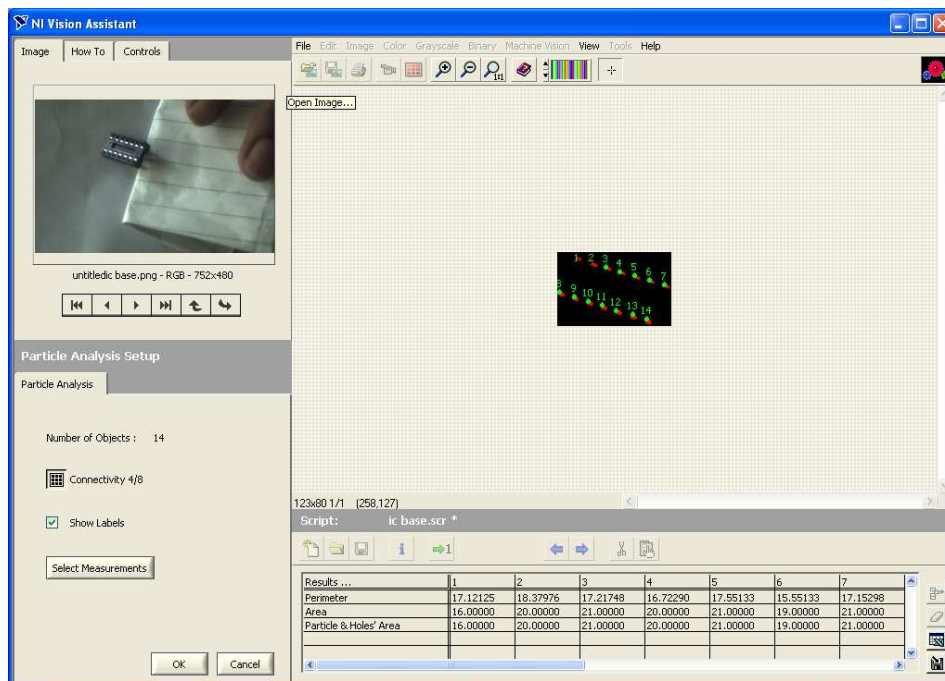


Figure 3.30 Particle analysis

### 3.4.6.5 Result

Object	Normalized Perimeter	Normalized Area	Normalized Particle & Holes' Area	Number of Holes
1	17.12125	16	0	0
2	18.37976	20	0	0
3	17.21748	21	0	0
4	16.7229	20	0	0
5	17.55133	21	0	0
6	15.55133	19	0	0
7	17.15298	21	0	0
8	18.80984	27	0	0
9	18.37976	26	0	0
10	18.37976	27	0	0
11	18.88363	30	0	0
12	20.38905	28	0	0
13	18.45355	29	0	0
14	22.0552	31	0	0

Table 3.8 Pin count of 14 pin IC base (Inclined)

### 3.4.6.6 Result 28 Pins

Object	Normalized Perimeter	Normalized Area	Normalized Particle & Holes' Area	Number of Holes
1	25.47754	47	47	0
2	24.37976	47	47	0
3	24.38905	42	42	0
4	22.32456	33	33	0
5	24.71361	43	43	0
6	23.98141	39	39	0
7	23.47754	39	39	0
8	20.80984	34	34	0
9	24.37976	44	44	0
10	24.37976	46	46	0
11	23.55133	42	42	0
12	23.47754	40	40	0
13	23.04746	33	33	0
14	21.88518	36	36	0
15	23.88518	37	37	0
16	23.22677	37	37	0
17	18.7229	25	25	0
18	22.7874	32	32	0
19	23.56062	34	34	0
20	16.32456	19	19	0

21	21.55133	30	30	0
22	20.0552	27	27	0
23	20.37976	30	30	0
24	20.0552	29	29	0
25	22	28	28	0
26	19.16228	22	22	0
27	19.94967	27	27	0
28	20.0552	28	28	0

*Table 3.9 Pin count of 28 pin IC*

### 3.4.6.7 Result 80 Pins

Object	Normalized Perimeter	Normalized Area	Normalized Particle & Holes' Area	Number of Holes
1	25.47754	47	47	0
2	24.37976	47	47	0
3	24.38905	42	42	0
4	22.32456	33	33	0
5	24.71361	43	43	0
6	23.98141	39	39	0
7	23.47754	39	39	0
8	20.80984	34	34	0
9	24.37976	44	44	0
10	24.37976	46	46	0
11	23.55133	42	42	0
12	23.47754	40	40	0
13	23.04746	33	33	0
14	21.88518	36	36	0
15	23.88518	37	37	0
16	23.22677	37	37	0
17	18.7229	25	25	0
18	22.7874	32	32	0
19	23.56062	34	34	0
20	16.32456	19	19	0
21	21.55133	30	30	0
22	20.0552	27	27	0
23	20.37976	30	30	0
24	20.0552	29	29	0
25	22	28	28	0
26	19.16228	22	22	0
27	19.94967	27	27	0
28	20.0552	28	28	0
29	20.32456	27	27	0

30	19.55133	29	29	0
31	21.62512	27	27	0
32	22.88363	35	35	0
33	21.47754	34	34	0
34	22.7229	33	33	0
35	19.55133	27	27	0
36	22.89292	28	28	0
37	19.21748	27	27	0
38	20.7229	30	30	0
39	19.80326	29	29	0
40	22.7229	31	31	0
41	20.0552	31	31	0
42	22.0552	39	39	0
43	22.7229	34	34	0
44	23.47754	39	39	0
45	21.21748	34	34	0
46	22.80984	37	37	0
47	22.0552	37	37	0
48	20.0552	29	29	0
49	17.88518	18	18	0
50	20.32456	22	22	0
51	20.38905	25	25	0
52	18.88363	23	23	0
53	20.0552	23	23	0
54	20.0552	27	27	0
55	19.55133	25	25	0
56	20.38905	25	25	0
57	64.31526	172	172	0
58	64.53429	190	190	0
59	64.31526	180	180	0
60	62.71361	183	183	0
61	61.87588	147	147	0
62	60.71361	176	176	0
63	61.54203	184	184	0
64	59.02887	165	165	0
65	60.63052	159	159	0
66	61.54203	170	170	0
67	63.54203	171	171	0
68	60.36116	166	166	0
69	60.71361	176	176	0
70	57.20818	157	157	0
71	58.97367	166	166	0
72	57.20818	167	167	0

73	59.80209	162	162	0
74	60.95508	185	185	0
75	57.47754	141	141	0
76	57.47754	148	148	0
77	56.63982	154	154	0
78	52.71361	129	129	0
79	53.03816	123	123	0
80	20.0552	28	28	0

*Table 3.10 Pin count of 80 pin IC*

### 3.4.6.8 Results 60 Pins

Object	Normalized Perimeter	Normalized Area	Normalized Particle & Holes' Area	Number of Holes
1	25.47754	47	47	0
2	24.37976	47	47	0
3	24.38905	42	42	0
4	22.32456	33	33	0
5	24.71361	43	43	0
6	23.98141	39	39	0
7	23.47754	39	39	0
8	20.80984	34	34	0
9	24.37976	44	44	0
10	24.37976	46	46	0
11	23.55133	42	42	0
12	23.47754	40	40	0
13	23.04746	33	33	0
14	21.88518	36	36	0
15	23.88518	37	37	0
16	23.22677	37	37	0
17	18.7229	25	25	0
18	22.7874	32	32	0
19	23.56062	34	34	0
20	16.32456	19	19	0
21	21.55133	30	30	0
22	20.0552	27	27	0
23	20.37976	30	30	0
24	20.0552	29	29	0
25	22	28	28	0
26	19.16228	22	22	0
27	19.94967	27	27	0
28	20.0552	28	28	0
29	20.32456	27	27	0
30	19.55133	29	29	0

31	21.62512	27	27	0
32	22.88363	35	35	0
33	21.47754	34	34	0
34	22.7229	33	33	0
35	19.55133	27	27	0
36	22.89292	28	28	0
37	19.21748	27	27	0
38	20.7229	30	30	0
39	19.80326	29	29	0
40	22.7229	31	31	0
41	16.46284	20	20	0
42	22.7229	39	39	0
43	22.64911	30	30	0
44	28.45355	51	51	0
45	23.55133	36	36	0
46	29.20818	55	55	0
47	15.56062	17	17	0
48	20.88363	32	32	0
49	20.0552	27	27	0
50	20.37976	30	30	0
51	20.0552	29	29	0
52	22.0555	28	28	0
53	19.16228	22	22	0
54	19.94967	27	27	0
55	20.0552	28	28	0
56	20.0552	31	31	0
57	56.63982	154	154	0
58	52.71361	129	129	0
59	53.03816	123	123	0
60	20.0552	28	28	0

*Table 3.11 Pin count of 60 pin IC*

### **3.5 Linking with LabVIEW**

Vision Assistant 7.1 creates a LabVIEW VI corresponding to the algorithm we prototype in Vision Assistant. Based on the options we select, the LabVIEW VI Creation Wizard creates a new VI that implements the image processing steps of the current script or of a saved script file. Only LabVIEW version 7.1 or above is compatible with VA 7.1. It does not support LabVIEW version below 7.1.

LabVIEW programs are called virtual instruments, or VIs, because their appearance and operation imitate physical instruments, such as oscilloscopes and multi-meters. LabVIEW contains a comprehensive set of tools for acquiring, analyzing,

displaying, and storing data, as well as tools to help we troubleshoot our code. In LabVIEW, we build a user interface, or front panel, with controls and indicators. Controls are knobs, push buttons, dials, and other input devices. Indicators are graphs, LEDs, and other displays. After we build the user interface, we add code using VIs and structures to control the front panel objects. The block diagram contains this code. Use LabVIEW to communicate with hardware such as data acquisition, vision, and motion control devices and GPIB, PXI, VXI, RS-232, and RS-485 instruments.

### 3.5.1 Front panel

The front panel is the user interface of a VI. We build the front panel by using controls and indicators, which are the interactive input and output terminals of the VI, respectively. Controls and indicators are located on the Controls palette. Controls are knobs, push buttons, dials, and other input devices. Controls simulate the input devices on a physical instrument and supply data to the block diagram of the VI. Indicators are graphs, LEDs, and other displays. Indicators simulate the output devices on a physical instrument and display data the block diagram acquires or generates.

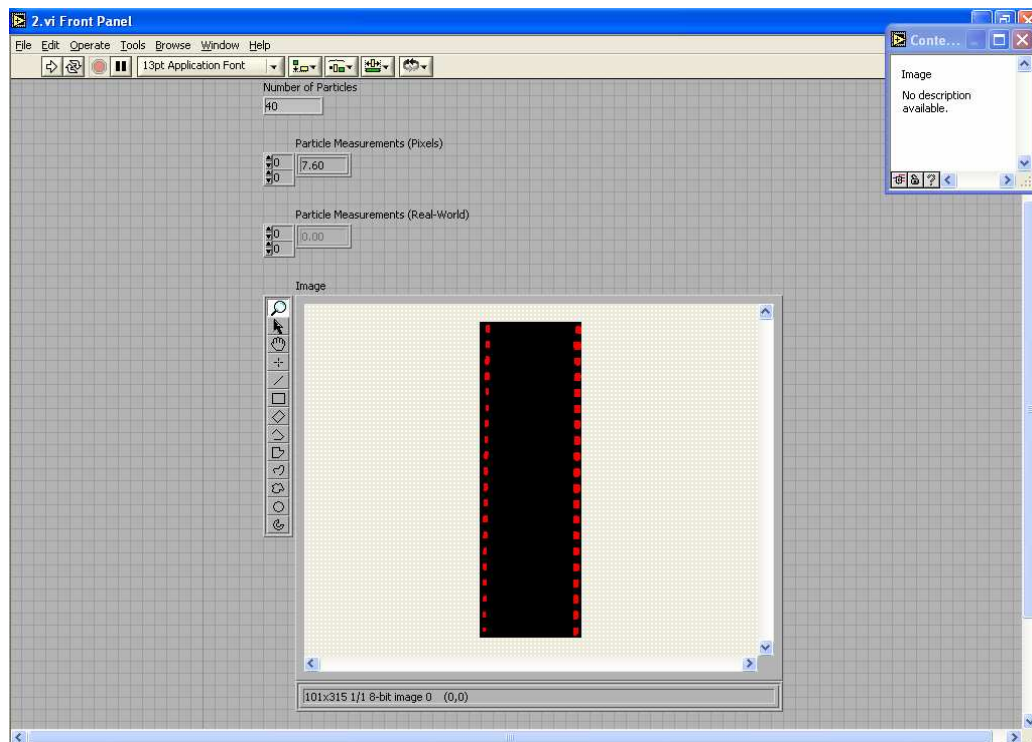


Figure 3.31 Front panel view

### 3.5.2 Block diagram

The block diagram contains the graphical source code for how the VI runs. Front panel objects appear as icon terminals on the block diagram. Wires connect control and indicator terminals to Express VIs. Data flows from controls to Express VIs, from Express VIs to Express VIs, and from Express VIs to indicators through the wires.

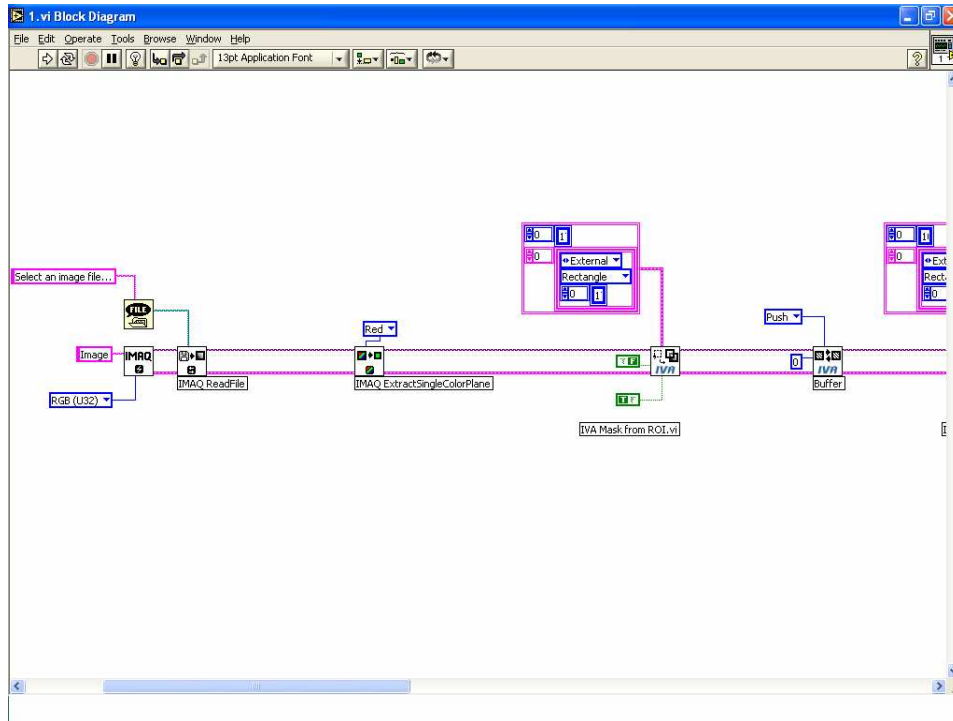


Figure 3.32 Block diagram part 1

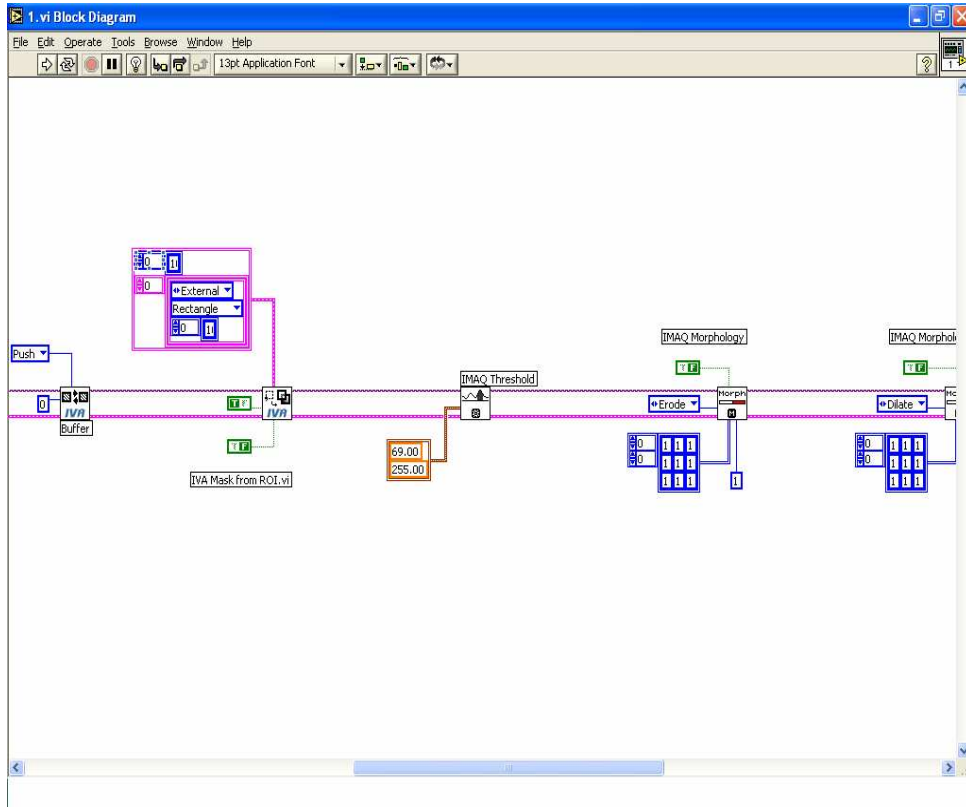


Figure 3.33 Block diagram part 2

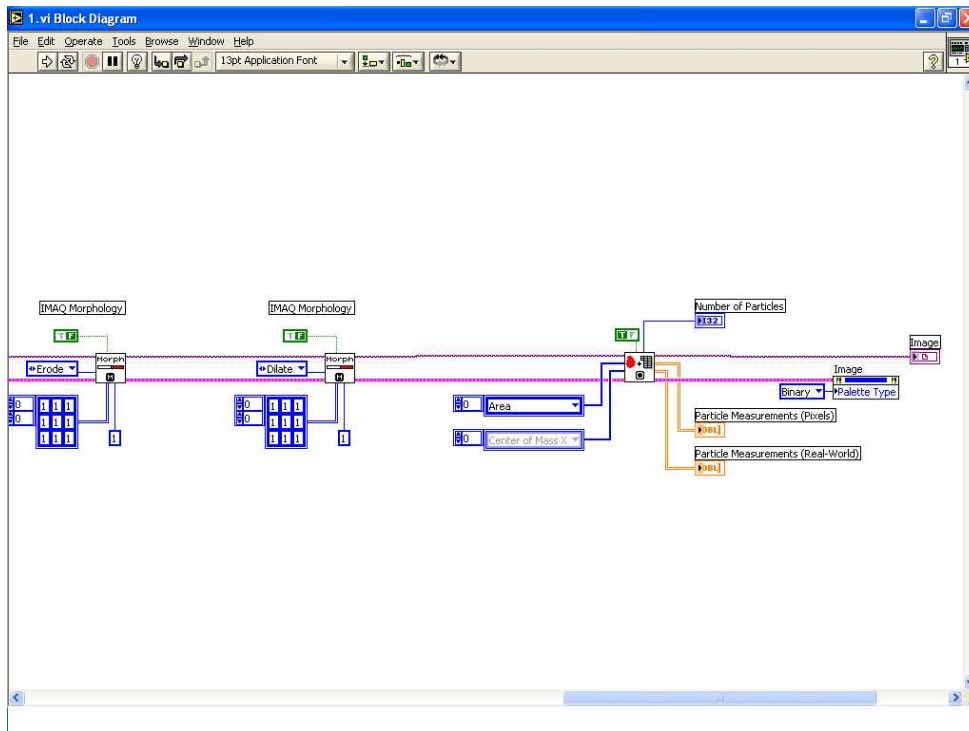


Figure 3.34 Block diagram part 3

### **3.6 Results and discussion**

The system shows the number of cells present in processed image. By counting number of pins present in an IC package such as 8, 14, 16, 40, 62, 80 pin, we can classify IC packages very fast as compared to manual counting. This algorithm is also adaptable to size and orientation of ICs. Position of IC may be horizontal, vertical or inclined the system is counting correct number of pins. Processing time is very less (4.63ms) as compared to manual counting of pins.

# Chapter 4

## Conclusions & future scope

---

Machine vision is an emerging area of technology in automation and control, wherein the images captured by camera are processed and analyzed in real time. Various machine vision applications include Agricultural robots, Under-water observations, Acrylic fiber production, Fire detection, Vehicle Trajectory approximation, Control of inverted pendulum, Weed sensing etc.

There are different software tools available for image processing and analysis. Machine vision assistant 7.1 is one of these tools and is useful for doing real time operations. IC pin counting is required in number of applications in the fields such as manufacturing and testing of electronic components. Manual counting of number of pins in this application is a tedious and time-consuming process. In this project Automatic IC pin counting device has been designed using principle of Machine Vision to make the process easy, fast and accurate.

A real time application requires live image capturing through on-line camera and instantaneous display for any corrective or control action to be taken. This project made use of images acquired and analyzed in real time.

Indeed imagination of the researcher is the only limit for the applications of Machine Vision. The day is not far when all visual inspection in industry and research will be replaced by Machine Vision. This Technology when interlaced with Robotics can realize all yester years' fictional characters. Thus to describe the future scope of Machine Vision in one word that would be appropriate is "IMMENSE".

## References

---

- [1] Nikku Saeed B., “*Introduction to robotics*”, Prentice Hall, India, 2005, pp. 248-306
- [2] David Eaton, Maurice Bobbitt, Melissa Reid, Garret Jarvis, Scott Frame, John Dowling, Mark Bloechl, Melody Mentzer Rachid Manseur “The Nautilus-Sub Autonomous Underwater Vehicle” *5th Annual AUVSI-UUV University of West Florida*, 2002
- [3] Paulo Heleno, Roger Davies, Bento A. Br ‘azio Correia Jo˜ao Dinis. “A Machine Vision Quality Control System for Industrial Acrylic Fibre Production”, *EURASIP Journal on Applied Signal Processing*, Hindawi Publishing Corporation, 2002, pp. 728–735
- [4] A.Milella, G. Reina, M. Foglia, and A. Gentile “Computer vision applications in agricultural robotics” *11th IEEE International Conference on Mechatronics and Machine Vision in Practice*, Macau, 2004
- [5] B.L. Steward and L. F. Tian, “Real-Time Machine Vision Weed-Sensing”, *ASAE Paper No 98 - 3033*, St. Joseph, MI: ASAE.
- [6] L. Wenzel, N. Vazquez, D. Nair, R. Jamal, “Computer Vision Based Inverted Pendulum” *Proc. of IEEE Conference on Instrumentation and Measurement Technology*, Maryland, USA, 2000, pp. 1319- 1323.
- [7] R. Fraile and S. J. Maybank, “Vehicle Trajectory Approximation and Classification”, *Proc. Brit. Machine Vision Conf.*, Southampton, U.K., 1998, pp. 832–840.

[8] Simon Y. Foo “A Machine Vision Approach to Detect and Categorize Hydrocarbon Fires in Aircraft Dry Bays and Engine Compartments” *IEEE transactions on industry applications*, Vol. 36, No. 2, March/April 2000, pp. 459-466.

[9] Manuals “*VA Tutorial*”, “*VA Concepts Manual*”, “*NI-IMAQ 1394 User Manual*”.

[10] [www.ni.com](http://www.ni.com)

[11] [http://sine.ni.com/apps/we/niepd\\_web\\_display.display\\_epd4?p\\_guid=C959340C4BF75058E034080020E74861&p\\_node=DZ52000&p\\_source=external](http://sine.ni.com/apps/we/niepd_web_display.display_epd4?p_guid=C959340C4BF75058E034080020E74861&p_node=DZ52000&p_source=external)

## Publications from this research work

---

1. Anurupa, Jainy Sachdeva, Mandeep Singh, Mandeep Singh, “Machine Vision: Principles and Applications”, *National Conference on “Computational Intelligence in Electrical Engineering (CIEE-05)”*, 18th -19th November 2005, pp 246-251.
2. Anurupa, Jainy Sachdeva, Mandeep Singh, Mandeep Singh, “Applications of Machine Vision”, *National Conference on “Advancement in Modeling & Simulation (AIMS’2006)”*, 20-21 January 2006, pp 320-325
3. Anurupa, Jainy Sachdeva, Mandeep Singh, Mandeep Singh, “Basic Principles of Machine Vision”, *National Conference on “Advancement in Modeling & Simulation (AIMS’2006)”*, 20-21 January 2006, pp 298-302