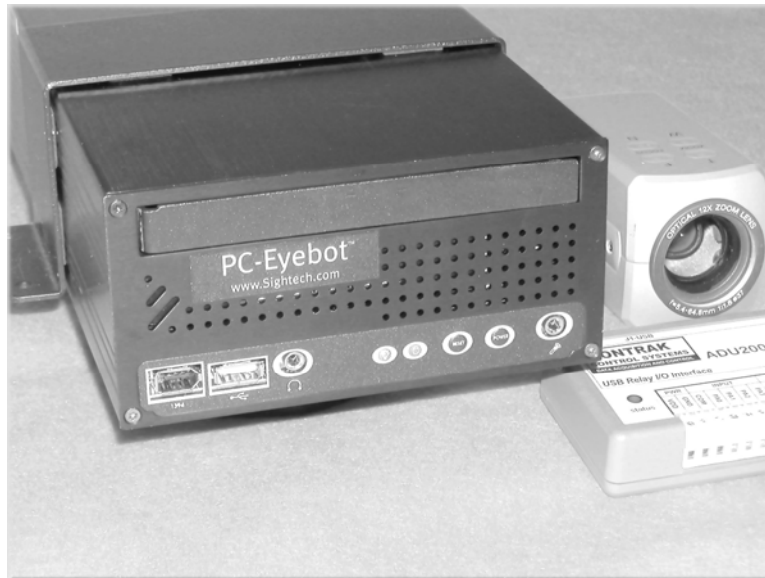


Sightech Vision Systems, Inc.

PC Eyebot



Users Manual

Published 2005

Sightech Vision Systems, Inc.
6580 Via del Oro
San Jose, CA 95126
Tel: 408.282.3770 Fax: 408.413-2600
Email: sales@Sightech.com
Web: www.Sightech.com

Table of Contents

Overview..... 4

Features.....	4
PC Eyebot Specifications	5
Typical System Configuration.....	6
Quick Start Guide	7
Installing the Software.....	7
Opening the Application.....	7
Running the software.....	7
Step 1: Erase	8
Step 2: View	8
Step 3: Learn.....	10
Step 4: Inspect	12
The PC Eyebot Interface.....	14
Describing the Interface.....	14
Drop-down File Name Window	14
Drop-down Area Selection Window	14
Final Part Decision Window	14
Action Buttons.....	15
IDLE	15
ERASE.....	15
VIEW	15
LEARN.....	15
FORGET.....	15
INSPECT	16
REC (COGNIZE)	16
Defining the Status Indicators	16
Indicators	16
Decision.....	17
Video	18
Mask	18
Feature	18
Position	20
Setting Up the System	21
Setting Up the System Components	21
Setting Up the Camera.....	22
Setting Up Lighting	22
Setting Up the Part.....	22
Inspection Modes.....	24
Choose the Application Mode	24
Shape-based.....	24

Color and Shape-based	24
Spectrum	25
Advanced	25
Setting the System Fixture Mode	26
None.....	26
Horiz_Coarse	26
Full-Fixtured.....	26
For Spectrum Mode Users.....	26
Examples of Different Fixture Settings.....	27
Explanation on Quick-Fixtured	29
Further Explanation on Nonfixtured	29
Further Explanation on Half-Fixtured.....	29
Further Explanation on Full-Fixtured.....	29
Running an Inspection	31
Configure the System	31
Start the Inspection Application	31
Train PC Eyebot to Inspect Parts.....	31
Using the Learn-Forget Mode	32
Train PC Eyebot to Recognize Parts	33
Supporting the Customer	35
Integration Support.....	35
Phone and Online Support.....	35
Feedback.....	35

Overview

PC Eyebot is a revolutionary device that takes advantage of the state-of-the-art techniques of machine vision. Gone are the headaches of programming a machine vision system. It produces results faster than traditional machine vision approaches. This device can also learn to recognize parts in a matter of seconds. This section contains the features and specifications.

Features

- Self-learning machine vision system—reduces setup time to a matter of minutes
- Cost effective—can be used on a wider range of applications
- Compact size allows PC Eyebot to be installed almost anywhere
- Inspect up to 1,800 parts per minute (PPM)
- Does not need a strobe or trigger input for many applications
- Offers modern low-noise high resolution digital cameras with or without trigger inputs
- Use to detect defects, to identify one set of parts from another, and identify objects within cluttered backgrounds
- Offers 3-state output decisions: PASS, FAIL, and MARGINAL. Each state can be assigned to an individual relay
- Recognize or inspect for defective parts with full X/Y movement and full rotation
- Full spectrum I/O: LAN, RS-232, USB, and 1394
- 40GB internal hard-disk data store – can save hundreds of project files for different parts, etc
- Up to 8 Relays to control notification and/or ejection mechanisms
- Offers many learning modes: Shape, Spectrum, Color combined with shape, and Coloration
- Offers many fixture modes: None, Horizontal, Vertical, Full (XY), and Radial.

PC-Eyebot™ Performance Qualification Specifications

The PC-Eyebot utilizes a massive feature-based process that processes up to 12 million features per second in order to perform high-quality inspections. Each feature may contain 100's of pixels of information.

Detection Modes of Operation:			
Execution Mode:	Training operation:	Execution operation:	Minimum of pixels needed for detection:
Absence/Presence Mode	Train on object/features to be detected, Forget on scenes where objects/features are absent	Detects presence and reports a score from 0 to 100	7x7 pixels minimum for object/feature to be detected.
Inspection Mode	Train on range of acceptable product	Detects defects and reports a score from 0 to 100	9x9 pixels for defect to be detected.

As an example, a 9x9 defect to be detected would need to be at least 9/1024 or roughly 1% of the field of view for XGA images. Similarly, an object/feature that needs to be detected must be roughly .75% of the field of view.

Camera image resolution capabilities available:								
Standard:	Mode:	Connector:	Vert Pixels:	Horiz Pixels:	Total # Pixels in Image:	Max Frame Rate of Camera:	Normal Non-Triggered Frame Rate:	Normal Triggered Frame Rate:
1/4 VGA	Digital	1394	240	320	76,800	60	60	24
NTSC	Analog	BNC	480	720	345,600	30	30	No Trigger
VGA	Digital	1394	480	640	307,200	60	30	18
XGA	Digital	1394	768	1024	786,432	15	15	10
SXGA	Digital	1394	1024	1200	1,228,800	7.5	7.5	7.5
UXGA	Digital	1394	1200	1600	1,920,000	3.75	3.75	3.75

General Capabilities:	
Description:	Clarification:
Simultaneous inspection Areas running at once.	Up to 64 allowed to be defined using Small memory. Up to 8 allowed using Medium memory. 2 allowed using Large memory.
Simultaneous cameras running at once.	Right now, one camera per PC-Eyebot
Area capabilities	Each Area may be separately defined, each with its own separate training memory and parameter settings.
Types of inspection:	Grayscale, Spectrum, Color, Coloration, and Texture
Feature sizes in pixels - tuned for certain types of inspection:	3x3, 5x5, 13x5, 9x9, 21x21, 27x27, 61x61, 69x25, 87x7, and more.

Sightech offers customized Product Inspection Validation Services where we perform laboratory tests on actual product and certify our equipment to operate at this specified level.PC Eyebot Specifications

Physical Specifications	
Dimensions	(WxHxD) 5.75” x 2.75” x 10”
Steel chassis	With aluminum panel
CPU unit	Detachable, easy maintenance and upgrade
Front bezel	NEMA 4/12 and IP65 compliant
Color	Black
Cooling	Fan
CPU Specifications	
Power Supply	120W ATX
Processor	High-end Pentium-4 or Pentium-M CPU
Mass Storage	30GB 2.5in HDD
I/O Ports	RS-232, parallel port (supports ECP/EPP, PS/2 keyboard interface, PS/2 mouse interface
Memory	512MB
Connectivity	Serial, 2x USB, 1394, LAN

Typical System Configuration

A typical system configuration consists of:

- PC Eyebot
- Digital Video Camera
- VGA/XVGA Monitor
- Lights
- Zoom Lens
- Cables and Power Supply
- USB Relay

Quick Start Guide

This section covers the following topics:

- Installing the software
- Opening the application
- Running the software

Installing the Software

The PC Eyebot software comes pre-installed at the factory. If for some reason you need to re-install it, please call SIGHTech for instructions.

Opening the Application

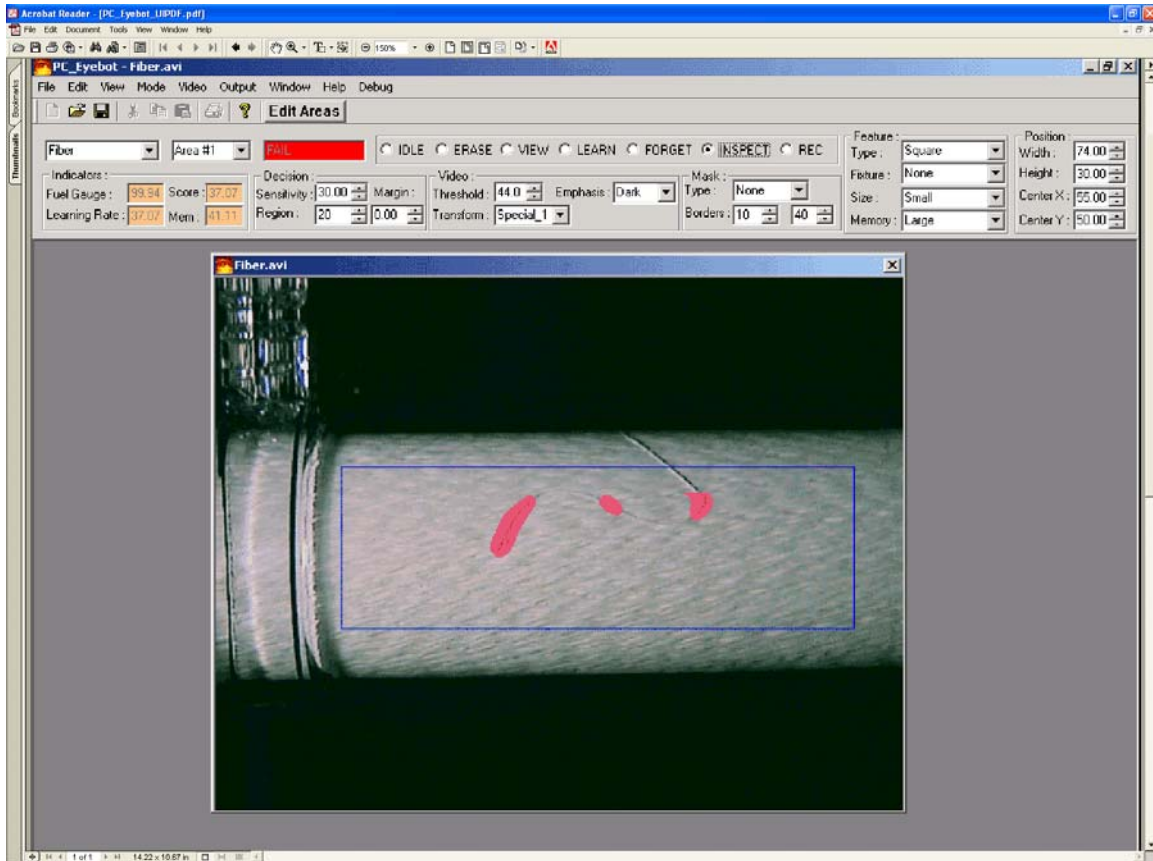
To open the application:

1. Click on Start Menu ->Program Files->PC Eyebot or double-click on the PC Eybot icon.
2. Observe the user interface as shown below with a running application file.

Running the software

Run the software if:

- Lighting is set up
- Camera is set up
- You are familiar with PC Eyebot's interface
- You have sample parts to inspect



You can train the PC Eyebot for inspection and perform the inspection with the following 4-step procedure.

Step 1: Erase

The first step to begin an inspection session is to do an ERASE:

1. Click the ERASE action button on the toolbar.

This will completely erase the information from any previous training that PC Eyebot has learned. ERASE clears the training you have done with LEARN.

Erase mode does *not* erase the system's settings, such as the window size, the output delay times, or the fixture setting.

Note! Be careful with ERASE because it will destroy all previous training!

Step 2: View

The second step to begin an inspection session is to do a View:

1. Click the VIEW action button on the toolbar.

2. Adjust the size of the area window and the image to your specifications.
3. Verify that the VIEW window is only looking at areas of the picture that are relevant to your application.
4. Set the Video Threshold on the toolbar (in the VIEW mode) to be as dark as possible and yet still able to image the defect.
5. Use the Video Threshold Up and Down buttons to adjust the Video Threshold in VIEW.

View depicts the “World According To PC Eyebot.” In a rectangle window on the monitor you see a threshold image or an edged image derived from the original image. The rectangle window is the area of inspection, the PC Eyebot’s field of view. The image inside the window is exactly what PC Eyebot sees, learns and inspects.

You can define multiple areas on the part for inspection using the EDIT AREAS button on the toolbar.

Note! The objective is to see the image of the features of the parts you want to learn and inspect in the VIEW mode. If you cannot see a difference between the good and bad parts in the VIEW mode, then PC Eyebot will be unable to see the difference.

What you should see on the screen in the View mode

The basic objective is to teach PC Eyebot the least amount of information it needs to do the job. Do not teach it things it does not need to know to do its job. For example, if you just want to see the cap of a bottle, do not make your VIEW look at the whole bottle, just focus on the top.

By minimizing the information PC Eyebot must learn, your training time will decrease and your accuracy will increase.

The image within the window is either the video image or the edged image (EDGE). In the Edge mode, most areas of the screen image should be solid black; the other areas represent the information of object parts. Some areas may be *glittering or showing a snowfall-like display*. These glittering areas, called **Sparkle**, are random change occurring in an image near the threshold value. Avoid too much of Sparkle because it quickly fills up PC Eyebot’s memory and makes learning more difficult. Make sure the areas are well defined and the part to inspect is clearly visible and features are distinguished. You just need to see the part edges or features, especially the features that make the part good or bad.

You can wipe out most Sparkle by properly adjusting the Video Threshold with the Up/Down buttons. Although, in trying to eliminate Sparkle, it is difficult to eliminate it completely. PC Eyebot can handle learning some Sparkle.

An optimal image in VIEW shows just enough detail to see the defect in the VIEW rectangle, but nothing more. The goal is to capture just enough detail to see the features you want PC Eyebot to learn and inspect. To achieve this, do the following experiment:

1. Place a good part or image of a good part while in VIEW mode and study the image.
2. Put a bad part under VIEW mode, and determine if you can tell the difference between a good part and bad part just by looking at the VIEW window.
3. Adjust the Video Threshold with the Up/Down buttons while in VIEW mode to bring out more details, if the answer is no. Consider changing lighting or zooming in (or out) of the image.
4. Continue darkening the image in the VIEW screen, if the answer is yes. This will simplify the amount of information PC Eyebot has to learn.
5. Repeat step 1 (Place a good part...) after making the VIEW darker, until you get an image where you can just barely see all the defects and features you want PC Eyebot to catch. You may have to put several bad parts under the VIEW screen to make sure that PC Eyebot is able to see all your defects.

SIGHTech Tip: Trying to get rid of Sparkle? If you have a part on a white background, try a black background instead, or vice versa. This may give PC Eyebot more contrast. Try adjusting the lighting.

The point of the preceding experiment is to show that you should not have PC Eyebot learn extraneous information. Just have it learn what is necessary to do the job. This will take some trial and error at first as you test to find the optimal Threshold setting in VIEW. But, it will pay off when you actually set it up.

Note: It is quite possible for you to see features that PC Eyebot cannot actually see when you are *not* in the VIEW mode. This is why it is important to pay careful attention to what you can see in the VIEW mode because this is what PC Eyebot actually processes.

Step 3: Learn

The third step to begin an inspection session is to do a LEARN:

1. Click the LEARN action button on the toolbar.

2. Turn on the conveyor belt to have your good parts go through the camera's view.
3. Verify that the status indicator Score is counting as the PC Eyebot is learning.
4. Observe that Score starts from 100 and will ideally progress to 0.
5. Set the Decision Sensitivity (DS) so that Score is below the DS when it is a good part, and above the DS when it is a bad part.

While in the “Learn” mode you will see dots where it is learning. The dots/Sparkle should go away once the learning is done.

When PC Eyebot starts learning, Score will have a large range. This is normal. It processes all the video in real-time. In a few seconds or minutes Score's range will begin to stabilize.

For example, say you get Score values that range from 5-25 when good parts go through. Therefore, you could set the DS at 30, so that when PC Eyebot sees something new or different, Score will go above the DS.

Note: PC Eyebot learns *everything* that goes through the VIEW rectangle. Even your hand if you stick it in the field of view! Learning is cumulative and PC Eyebot is not just taking a few snapshots. There is no frame grabber in PC Eyebot. It is processing every video frame in real-time and making over 13 million microdecisions per second, which lead to one macrodecision.

The lower the Score, the more PC Eyebot is trained.

When is PC Eyebot done learning?

However, in most applications if Score is consistently below 20, then PC Eyebot has learned enough.

How big of a defect am I trying to detect?

If it is a gross defect, then Score may not even have to go below 50. Why? Because, you will get such a large dynamic range in Score between a good part and bad part. Therefore you can easily set your Decision Sensitivity with plenty of margin. However, if you are trying to see very subtle defects, then you may want to see Score staying consistently in the 10s before stopping learning.

Note: Only acceptable parts should be going through the camera view while PC Eyebot is in LEARN (unless you want it to learn bad ones). If a bad part should go through, some

of its features will be assimilated into memory and these features will then pass in the future when PC Eyebot inspects.

How long does learning take?

The more you features you demand PC Eyebot to learn, the longer it will take to learn them. For example, the more positions and orientations you make PC Eyebot learn, the longer the process will take.

PC Eyebot will usually learn stationary parts in a few seconds. Learning complex parts, such as a rotating dollar bill, might take 15 minutes or more. If Score is above 50 and does not fall and/or is stuck in a zone, that means PC Eyebot is still learning the features in the image. However, if it does move after a couple of minutes, it means one of the following:

- There is too much sparkle interfering on the image (see the VIEW section on how to correct this).
- The part and/or its movement are too complex.
- The part is not placed at the correct distance from the camera, or the camera is out of focus.
- PC Eyebot is out of memory (this is rare).

Note: PC Eyebot can still perform quite effectively even with partially learned parts. It is likely that you do not have to wait for Score to hit the 90s for PC Eyebot to accomplish your application. If the training has taken it over 70, that is usually sufficient for most applications.

Step 4: Inspect

The fourth step to begin an inspection session is to do an INSPECT:

1. Click the INSPECT action button on the toolbar.
2. Use the Up and Down buttons to set the Decision Sensitivity value slightly above the good part Score range.
3. Test PC Eyebot by placing a defective part on screen and observing if the Score goes above the Decision Sensitivity.
4. Readjust the Decision Sensitivity with the Up/Down buttons if the Score of a defect part does not cross with a significant margin.

5. Observe Score goes higher than Decision Sensitivity value when a defect appears, and fall below it when it is a good part.
6. Return to step 3 (LEARN) if this is not possible and train PC Eyebot some more.

In this mode, PC Eyebot will stop learning and start inspecting. The outputs are disabled in INSPECT mode; if you want to activate the external outputs turn PC Eyebot to RUN.

Under INSPECT mode PC Eyebot will Score all the parts based on what it has learned during the LEARN phase. The good parts (the parts you have taught PC Eyebot) obtain lower scores and the parts with defects are scored high.

While in INSPECT mode you can set and adjust the Decision Sensitivity. The Decision Sensitivity represents the level of strictness you want to set.

You can adjust the Decision Sensitivity using the Up/Down buttons when PC Eyebot is on either the LEARN, INSPECT, or IGNORE modes. You can make PC Eyebot stricter in its decisions by moving the Decision Sensitivity value down. To make PC Eyebot more lenient in its decisions push the Down button.

For example, after monitoring the process for a couple of minutes under INSPECT, you have observed that the good parts Score ranges from 5 to 20. In this case, 20 represents the marginally good parts and 5 represents the best parts. Therefore, you would set the Decision Sensitivity around 25 (to give a little margin of error).

PC Eyebot's learning is cumulative; therefore, returning to learn will not erase previous learning, but rather will build on the previous learning experience. Under INSPECT mode, the outputs are activated, including the Beeper.

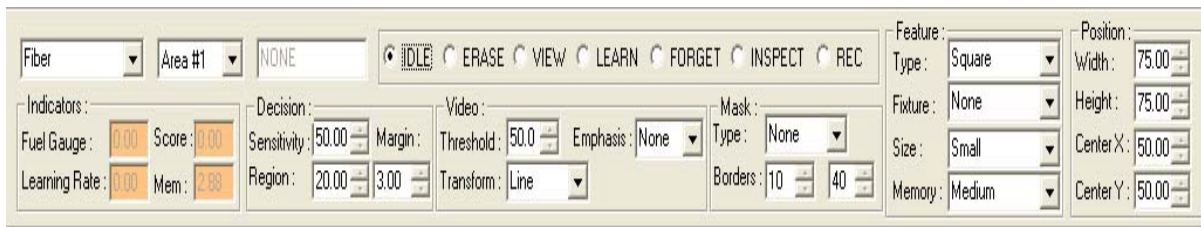
The PC Eyebot Interface

Describing the PC_Eyebot interface menu items includes:

- Describing the Interface
- Defining the Status Indicators

Describing the Interface

When you open PC_Eyebot, you see the menu bar and the main window. Locate the midlevel toolbar.



The midlevel toolbar consists of:

- Drop-down File Name Window
- Drop-down Area Selection Window
- Final Part Decision Window
- Action Buttons

Drop-down File Name Window

This window (unnamed) is located on the midlevel toolbar on the far left. It indicates a user application that is currently open and running.

Drop-down Area Selection Window

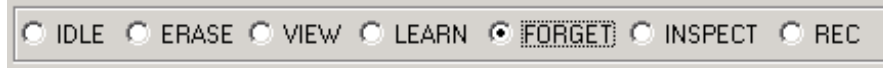
This second box indicates the area that is currently active. The user can manually select the area name from the drop-down window list. Or, the user can graphically select the area on the video screen using the mouse. Alternatively, all areas can be made active.

Final Part Decision Window

This window shows the final part decision for the current part. The results indicate a Pass, Fail, or Marginal decision.

Action Buttons

The action buttons include the following click-type:



IDLE

Place the system in an idle mode to do nothing. Video will be streamed and shown on the screen, but no processing or inspection is done.

ERASE

This button erases all “learnt” memory about the part. Ideally “Erase” should be used before user learns new part. Warning: Erase will delete all features about the part in memory. It is not possible to regain the erased memory.

VIEW

The View button enables a graphical view of the video with areas and features. No processing or inspection is done in the view mode. View mode is similar to IDLE mode, except more data is shown on the video window.

In VIEW mode, you will see most of the video, but the viewing area will in the EDGE mode, where you will see only the processed edges in the video. This EDGE display happens while you are “Feature Mode: Shape”. In the other color or spectrum mode, you will not see these feature edges. For example, in color mode, it will show all video and colors, since edges are not learned and are not part of the processing.

LEARN

This button enables the system to “LEARN” the part. While learning you will see “orange” speckles. These are new feature areas or locations the system is learning. Initially you will see these learning dots in many places. As time goes on (for example, a few seconds), the dots will decrease. This indicates the system has learned more and less remains to be learned, unless the part is changing, rotating, or moving. Then, more learning will occur. Learning is a critical step in this process.

FORGET

This feature enables system to “unlearn” or forget what is learned in the displayed areas. The user can learn and selectively “forget” or “unlearn” certain things. This feature is unique in two ways. First, it provides tools to erase certain parts of learning the part. It is like doing a selective ERASE.

FORGET is also used for “Presence vs. Absence” detection. When you use this feature, the system will display “Present and Absent” results. The system will not do pass-fail detection. For example, to detect a part, the system learns the part with the scene. Then remove the part and select FORGET. This will enable the system to forget the background, but retain the part in memory. Then in the REC mode, the system will tell you if a part is present or absent.

INSPECT

This button enables the system to inspect the part for defects.

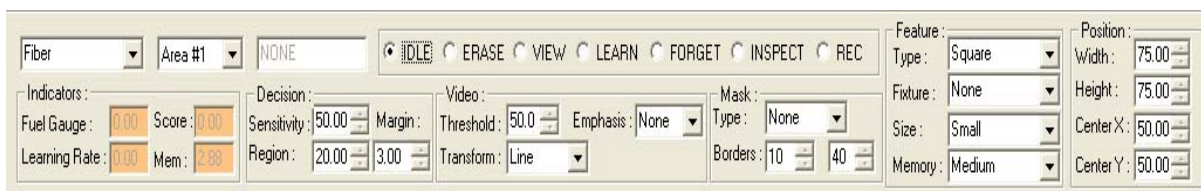
REC (COGNIZE)

This button is used for “Presence vs. Absence” detection.

Defining the Status Indicators

The Status indicators are located at the bottom of the tool bar. They are grouped into four classifications:

- Indicators
- Decision
- Video
- Mask



Indicators

The indicators include:

- Fuel Gauge
- Learning Range
- Score
- Mem (Memory)

Fuel Gauge

This button indicates how much “resource” fuel is left on the machine. Resources include memory, CPU etc. This number is expressed in percentage and ranges from 0 – 100 indicating what is still available.

Learning Rate

This button shows how much the system is learning. This number will stop and stabilize when the system is finished learning. This number is expressed in percentage and ranges from 0 – 100.

Score

This is the final part inspection result. A Score of certain values passes; below a certain value a part will fail. This number indicates the score for the current part. See *Running a Sample File, Step Three: Learn*, for more score information.

Mem (Memory)

This button displays the computer memory left on the machine. This number is expressed in percentage and ranges from 0 – 100.

Decision

Decision includes:

- Sensitivity
- Region
- Margin

Sensitivity

Sensitivity displays a number the user specifies for a part “Pass-Fail” threshold. There are two levels of decision-making. One level is regional and the other is global. First, a regional decision is made at a lower level. Then, a higher-level threshold is made globally on the entire part. This value is the global threshold.

Region

Region displays a number the user specifies for a part “Pass-Fail” threshold at the regional/local level. There are two levels of decision-making. One is regional and the other one is global. First, a regional decision is made at a lower level. Then, a higher-level threshold is made globally on the entire part. This value is the regional threshold.

Margin

Margin displays a number the user specifies to indicate a “Marginal” part. A part can be classified as marginal, based on this margin value.

Video

Video parameters include

- Threshold
- Transform
- Emphasis

Threshold

Threshold is used for “shape” inspection. The value set by the user indicates how much of the part’s shape is seen or extracted. The higher the threshold value, the less features are seen. The lower the number, more noise and smaller edges are seen. The optimum number is based on each part and established by the user.

Transform

This indicates the “part” filter. The filter defines how the feature edges are extracted. Mathematically, the user can use horizontal, vertical, or angular features etc.

Emphasis

The user can emphasize certain types of visual characteristics such as bright, dark, red, green, or blue areas. This can help learn to better learn some specific things.

Mask

Mask includes:

- Type
- Border

Type

This defines the type of mask to use to look at an area. An area can have no masks. In some cases, special masks like: Circular, Rectangular masks can help focus on only certain regions in an area.

Border

This defines the region for the mask selected above

Feature

Features include the following drop-down windows:

- Type
- Fixture
- Size
- Memory

Type

This button defines the type of “feature” to be used to learn the parts. Feature shapes can be square, rectangle, or circle etc.

The feature type can also be shape, or color based. In shape-based, it will only learn the shape of the feature or part. In a color-based, it will only learn the color ranges. There is a hybrid mode, where it will learn some shape and some color. There are three such hybrid modes:

- Color_10: 10% color and 90% shape
- Color_50: 50% color and 50% shape
- Color_90: 90% color and 10% shape

Fixture

Fixture defines how the part is fixtured. The choices are:

- Nonfixtured: The part can move anywhere and inspection should happen all over
- Half-Fixtured: The part moves horizontally, meaning the features are locked vertically, but can move horizontally
- Fixtured: The part is fixed and all features should occur in the same place

Size

This drop-down window defines the size of the feature:

- Small
- Medium
- Large

Memory

This defines the memory model required:

- Small
- Medium
- Large

Position

All the position parameters are applied on the current area selected. This is primarily to adjust and manipulate the area size and position. Position includes the following drop-down windows:

- Width
- Height
- Center X
- Center Y

Width

Adjusts the width of the current area rectangle.

Height

Adjusts the height of the current area rectangle.

Center X

Adjusts the position (X-center) of the current area rectangle.

Center Y

Adjust the position (Y-center) of the current area rectangle.

Setting Up the System

Prepare the PC Eyebot for operation with the following components:

- PC Eyebot, power supply, keyboard and mouse
- Digital 1394 (Firewire) camera with a steady mount
- 1394 (Firewire) cables to connect PC Eyebot to the camera
- VGA Monitor and VGA cable (needed only during training and is helpful to monitor the process)

Set up the PC Eyebot by doing the following:

1. Connect the digital 1394 video cable from the camera to the 1394 connector on the front of PC Eyebot.
2. Connect the input cable from the monitor to the monitor connector at the back of PC Eyebot.
3. Connect the keyboard and mouse cables to the corresponding connectors at the back of PC Eyebot.
4. Plug the PC Eyebot power supply into the back of the PC Eyebot.
5. Press the POWER button at the front of PC Eyebot to start up system

Setting Up the System Components

The components to set up include:

- Camera
- Lights
- Part

Setting Up the Camera

Mount the camera and make sure it is stable. Use an adjustable mount to make experimentation easier. Then do the following:

- Start the PC Eyebot application (see *Opening the Application*).
- Verify that the camera image is displayed on the monitor

Focus on the area to inspect. For example, to verify that a car has a door handle, make the door handle image fill up the video screen.

Setting Up Lighting

In machine vision, lighting is frequently 90% of the battle (some would say 99%!). SIGHTech has designed the PC Eyebot to make this task relatively easy. Nevertheless, there are some applications that are challenging and demand attention to lighting conditions. Here are a few tips that may help you use lighting to your advantage.

- Try different lights (halogen, incandescent, fluorescent, etc.)
- Try different intensities (by dimming or by approaching the part)
- Try different angles (to highlight grooves or eliminate sparkle)
- Try to alter the lighting by moving your hand over the part. When you find the optimum spot, place a permanent part to duplicate the soft shadow your hand created
- Work with a machine vision integrator that specializes in lighting

Setting Up the Part

Make sure the parts are oriented and placed so that the area of inspection is visible by the PC Eyebot. To check the placement, use the monitor.

Minimize complex backgrounds, especially changing ones. Use a black or white backdrop if possible. Background interference can be managed by using a lens with a small depth-of-field. Carefully focus on the part while defocusing (blurring) the background.

If the parts are moving, try to fixture the movement in a consistent manner across the screen. Fixtured stationary parts are the easiest to inspect. Ideally, train PC Eyebot to LEARN and inspect the parts while they are stationary. The more aspects of a part PC Eyebot learns, the less discriminating it will be when inspecting. Ultimately, if PC Eyebot learns too many aspects, it will only catch gross defects.

Inspection Modes

PC Eyebot offers 4 categories of inspection modes:

- Shape-based
- Color and Shape-based
- Spectrum
- Advanced.

Choose the Application Mode

Choose the mode that meets the needs of your applications.

Shape-based

This mode can include the following shapes:

- Rectangle
- Square
- Circular
- Burst
- Wide

This mode allows PC Eyebot to learn and inspect shapes. PC Eyebot sees the world as millions of small shape-based features/edges. It sounds an alert when it sees new features that it was not taught.

Note: Shape-based modes do not learn colors.

Color and Shape-based

This mode include the following:

- Color_10
- Color_50
- Color_90

This mode combines the Shape and Spectrum capabilities of PC Eyebot. It not only learns shapes, but also takes into account the color of the object.

Color_10 combines about 10% color information with 90% shape-based information.

Color_50 combines color and shape information 50/50.

Color_90 combines 90% color information with 10% shape information.

Spectrum

This mode include the following:

- Spectrum
- Coloration

This mode learns color spectrums. Use Spectrum mode to see a color difference and don't care about the shape of product.

Use Coloration mode to learn and inspect for small color graduations

Note: Spectrum and Coloration modes do not learn shapes

Advanced

This mode include the following:

- Behavior
- Special_1
- Special_2

The advanced modes are not currently implemented.

Setting the System Fixture Mode

PC Eyebot has three option settings to designate the fixture mode for the parts to be inspected:

- None
- Horiz_Coarse
- Full-Fixtured

None

None makes PC Eyebot relatively insensitive to X/Y movement of a part. This setting is best when there isn't much control over the parts during the inspection process.

Horiz_Coarse

Horiz_Coarse makes PC Eyebot relatively insensitive to right to left movement (on the screen). Otherwise, it is very sensitive to up and down movement (on the screen). This is useful for the PC Eyebot to detect minor up and down shifts (for example, seeing a bottle cap coming loose).

Full-Fixtured

Full Fixtured makes PC Eyebot very sensitive to both up and down and right to left movement. This is useful when you have a rigidly fixed part on an X/Y axis and it is important to detect minor changes.

Use Half-Fixtured when parts are moving from right to left or from left to right on the monitor screen.

Use Full Fixtured when parts are indexed into position. When using Full-Fixtured you will usually interrogate the relay output only when the part has stopped moving. Or, use the Wake Up pin to signal PC Eyebot to start inspecting.

Most applications can and should be solved with Nonfixtured. Some processes that seem very controlled are not, so Nonfixtured works well.

For Spectrum Mode Users

In Nonfixtured, Spectrum mode learns all the acceptable color spectrums and applies them everywhere. It does not inspect shapes. Use Nonfixtured only when you want to detect new colors. For example, if Spectrum mode learns a US flag, it will pass a French flag since it has the same three colors.



















Nonfixtured is useful in food processing. For example, where you want to learn the acceptable colors of French fries or tomatoes to fail when the colors start to change. Shape of the part is frequently irrelevant in food processing, but color can be critical.

Only Full-Fixtured detects missing or misplaced colors. However, it depends on very tight fixturing of the part.

Examples of Different Fixture Settings

To help you decide what method is best for your application, look at Table 1. In the examples, assume you are a manufacturer of smiling faces. In all the cases, PC Eyebot is trained to learn the smiling face in the same X/Y position. It sees about 20 samples of the face in the same position. Here are the results, depending on the setting on the PC Eyebot.

Table 1. Examples on Different Fixture Settings

Fixture Setting	Good Parts	PC Eyebot Will Pass	PC Eyebot Will Fail
Nonfixtured			
Use this setting when part is not fixtured on an X- or Y-axis. Especially useful when there is some rotation. PC Eyebot can learn parts in various orientations.		Passes because it has not rotated and has no new features. Even though it is not positioned precisely on an X and Y-axis, it does not matter because the setting is X and Y Nonfixtured.	Fails because it is either rotated or has a mark across it. Had the part been learned with +/- 20 degrees of rotation, then the top one would have passed, but it would still fail if it were at a 90° angle, for example.
Nonfixtured			
Half-Fixtured			
Use this setting when the part is fixtured on the X-axis, but not the Y-axis. Useful when parts are moving across the screen from the right to the left.		Passes because it remains on the Y-axis. Moving up or down does not affect it because it is Nonfixtured on the Y-axis.	Fails because either it has shifted off the Y-axis (above), or a defect has appeared (below).
Half-Fixtured			
Full-Fixtured			
Use this setting when part is <i>precisely</i> fixtured on X-axis & Y-axis. Useful when parts are extremely controlled and you want to detect subtle changes. It is rare that a process has enough control to use this mode.		Passes because it remains on the X-axis. Moving up or down does not affect it because it is Nonfixtured on the Y-axis.	Fails because it moved (above), or a small defect has appeared (below, it's the dot above the mouth).
Full-Fixtured			

Explanation on Quick-Fixtured

When PC Eyebot is Quick-Fixtured it is able to LEARN parts at a faster rate, however it loses its ability to detect small defects. Quick-Fixtured is useful for inspection processes containing parts with gross defects.

SIGHTech Tip: The majority of SIGHTech's customers use their PC Eyebot's in Nonfixtured mode.

Further Explanation on Nonfixtured

When PC Eyebot is Nonfixtured, features are classified and saved into memory. This is without regard to the physical location of the shape in the video image. If a part is learned in the upper right-hand corner, that same part may pass in the lower left-hand corner. We say that it *may* pass, because there may be subtle lighting differences from one corner to another corner of a camera view. This may make the part appear different to the digital preprocessor. Nonfixtured is useful since the position of an inspection part may be difficult or impossible to orient.

Further Explanation on Half-Fixtured

When PC Eyebot is Half-fixtured, the X-coordinate corresponding to the vertical position is stored with the feature. This mode is typically used when parts are moving from side to side through the video image, like bottles on a labeling machine. A feature that is learned on the left side of the video image will pass inspection on the right side. But, if learned on the bottom of the image, will not pass inspection at the top.

Further Explanation on Full-Fixtured

In Full-Fixtured mode, the features of a part and its position are combined and saved in memory. A part that is learned in the upper right hand corner will not pass in the lower left hand corner.

You may notice that a Full-fixtured application learns faster than a Nonfixtured application, *if the part is stationary*. For this, there are several reasons. One reason is that Nonfixtured mode learns a higher complexity of feature details in order to properly identify it across the entire video image. A higher feature complexity takes more time to learn.

Another reason that Full-Fixtured applications may learn faster is that they are less susceptible to Video Sparkle. Video Sparkle is small areas in the image that cannot be resolved with the video preprocessor. Consequently, it tends to generate a set of nearly random features. In a Nonfixtured application, these random features will degrade the performance throughout the video image.

In Full-Fixtured mode these random features will be contained to that small area. This occurs, since the random features will have position information attached to it. A random feature learned in the upper right hand corner, for example, does not affect the performance in the lower left hand corner.

The final reason a Full-Fixtured application generally learns faster than a Nonfixtured application is that Full-Fixtured applications are generally better designed. Usually, the amount of image variability is reduced. This occurs because the inspection part is brought into a fixed and known position before learning or inspecting takes place.

Running an Inspection

This section gives a quick overview of placing the PC Eyebot into operation on a quick-start basis to run an inspection.

Configure the System

To configure the system, do the following:

- Connect the PC Eyebot to the VGA monitor using a VGA cable
- Power both the VGA monitor and PC Eyebot with 110 VAC, using supplied AC-DC converters (wall packs)
- Connect the digital 1394 (Fire wire) camera using a 1394 cable.

Note: The PC Eyebot powers the digital camera via the 1394 cable

Start the Inspection Application

To start the inspection application:

1. Turn On the PC Eyebot. Check that the blue power light is on.
2. Start the PC Eyebot application.
3. Click File.
4. Click Open Digital Camera.
5. Type in name of project. PC Eyebot will setup a default work area.
6. You can now change size and/or location of work area (processing window):
Position → Width/Height/Center X/Center Y.

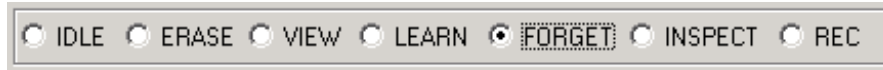
Train PC Eyebot to Inspect Parts

Place a good part in the camera's field of view. Do the following:

1. Click on Erase. After confirmation, memory is erased.
2. Click on View. Increase or decrease Video → Threshold so that you can clearly see the part and the details you want to inspect. The less complicated the resulting

image, the sooner PC Eyebot will learn the part.

3. Click on Learn (see the portion of the advanced tool bar below). This will start the learning process. Small orange dots on the image indicate areas of learning. These small orange specks are called “Hits.” As learning progresses, the number of these hits will decrease dramatically. If they do not, then the setup/lighting probably needs to be examined and improved.



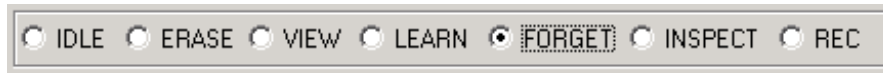
4. Observe the value in the Indicators→Score box. Continue training PC Eyebot until Score has dropped to 10 or less. If the part is stationary, this may take only a few seconds. If it is moving, it may take 1-5 minutes. Training should duplicate the conditions under which PC Eyebot will inspect. If there will be vibrations, tap the camera to simulate vibrations or other imprecisions.
5. Click on Inspect. The PC Eyebot is now actively inspecting.
6. Place a defective part in the field of view of the camera and check to see if the (Indicators→)Score increases.
7. Adjust the Decision→Sensitivity to make sure that good parts are passing and bad ones are failing. If this is not possible, make PC Eyebot Learn more examples of good parts and try to inspect again.
8. Return to step 4 and repeat until PC Eyebot is able to differentiate between good and bad parts.
9. When PC Eyebot is performing well in INSPECT, you may easily configure relays to enforce the decision: Output→Relays→Settings.

Using the Learn-Forget Mode

Most PC Eyebot users will use the inspection capacity of PC Eyebot. This means that PC Eyebot learns a process and whenever there is a deviation from what it learned it will tell you.

However, a few users may want to use the recognition capacity of PC Eyebot. Use this mode when there is a known universe of parts or defects that you are looking to detect. This mode is called Present/Absent because it detects the presence or absence of what it was trained to look. The goal is to LEARN PC Eyebot on the part(s) you want it to recognize and then FORGET (see the portion of the advanced tool bar below) the part(s) you do not want it to recognize.

PC Eyebot will display Present whenever it recognizes the part(s) it learned. PC Eyebot displays Absent when it sees the part(s) it was taught to FORGET.



For example, let us assume you make three types of soft drinks (Cola, Root Beer, and Lemon/Lime). You want PC Eyebot to make sure that you are only make one type of can and that a wrong one does not slip in. There are two ways to solve this problem using FORGET:

- Train it to LEARN the two cans that are “bad” and FORGET the one that it is “good.” PC Eyebot will send a signal whenever it sees one of the two “bad” cans. Whenever a “good” can passes, PC Eyebot will not send a signal.
- Another way is to train it to LEARN the “good” can and FORGET the two “bad” cans. Again, PC Eyebot will send a signal whenever it recognizes the can it learned (the “good” one in this case). It will not send out a signal on the ones that it was taught to ignore (the “bad” ones).

Most manufacturers would prefer using the first method because the result is more intuitive (that is, a signal on a reject). However, either method is effective.

Once you start ignoring, you cannot return to LEARN. If there is something you want PC Eyebot to learn about the part you are training it to recognize, then you should ERASE and start over. The key to effective training when using FORGET is to do as much learning about a good part at the beginning and then ignore all the differences. You can return to FORGET as many times as you need.

Train PC Eyebot to Recognize Parts

Follow steps 1-9 (under *Training PC Eyebot to Inspect Parts*) above. PC Eyebot can learn more than one part. To train:

1. Click on FORGET.
2. Show the part(s) you want PC Eyebot to ignore on the screen.
3. Press Yes. This will start the forgetting process. To stop or pause the forgetting process, press the No button. The Score will drop as PC Eyebot learns to forget the part(s).
4. Click on REC(OGNIZE).
5. Place the first part (the learned part) in the field of view of the camera.

6. Check to see if the Score rises above the Decision Threshold.
7. Verify that the Score drops by putting one of the forgotten parts in front of the camera.
8. Start again from step 2, if the Score does not drop (in step 7) until PC Eyebot can differentiate between the learned part(s) and the forgotten part(s).
9. Start over and change your setup, if PC Eyebot is unable to differentiate.

Supporting the Customer

Supporting the customer consists of:

- Integration Support
- Phone and Online Support
- Feedback

Integration Support

If you need assistance in integrating PC Eyebot with your needs, please contact SIGHTech. We can either help you directly or point you to a SIGHTech Approved integrator or reseller.

Phone and Online Support

For telephone support, please call 408.282.3770.

For online support, please visit SIGHTech's web page at www.SIGHTech.com, and/or write to Support@SIGHTech.com. If you purchased PC Eyebot through a SIGHTech reseller, please contact the reseller.

Feedback

SIGHTech would like to know what you think of PC Eyebot and/or this manual. Please [contact SIGHTech](#) on your comments. We take your feedback seriously and like to know how we can help you.