Sherlock™

machine vision software

User's Reference Manual

403-00007-00

For Software Version 7.3.1.0

TELEDYNE IMAGING
Everywhereyoulook™

Document Number 403-00007-00
November 17, 2020
Software version 7.3.0.0

NOTICE OF PROPRIETARY RIGHTS

This manual and the related hardware and software are confidential trade secrets and the property of Teledyne Imaging and its licensors. Use, examination, reproduction, copying, transfer and/or disclosure to others of all or any part of this manual and the related documentation are prohibited except with the express written consent of Teledyne Imaging.

The information in this document is subject to change without notice. Teledyne Imaging makes no representations or warranties with respect to the contents of this manual and specifically disclaims any implied warranties of merchantability or fitness for a particular purpose. Teledyne Imaging assumes no responsibility for errors or omissions in this document.

Teledyne Digital Imaging US, Inc.
Information: TDI_Sales.ipd@teledyne.com
Support: TDI_Support.ipd@teledyne.com
Web: http://www.teledynedalsa.com/visionsystems

700 Technology Park Drive
Billerica, MA, USA 01821
Tel 1.978.670.2002 **Fax** 1.978.670.2010

Preface

Congratulations on your purchase of Sherlock™ machine vision software! OEMs, System Integrators, and End-Users of machine vision find the rapid prototyping ability of Sherlock greatly reduces development time while speeding time-to-market. Some examples of what you can do from within Sherlock are:

- Graphically organize your entire machine vision strategy
- Interactively experiment with all functions of the vision processing library
- Confirm run-time execution performance
- Fine-tune variables and parameters
- Create custom vision algorithms
- Plot statistical data (SPC)
- Set or Read discrete I/O signals

Sherlock also provides Monitor, Reporter, and Watch utilities for gaining insight into vision task program flow, internal/external events, internal variables, or other parameters as you run your machine vision application.

When you have proven the application performance from within Sherlock and are prepared to deploy your application, options are available to you:

- Run the application directly from the Sherlock software environment. This deployment frees users from the task of programming, and allows operator interface with the vision application through the Sherlock graphical user interface.
- Run Sherlock programmatically through OCX using a VisualBasic or Visual C++ front end.

Regardless of how you deploy your Sherlock machine vision application, you can rest assured that you’ve built your application on a highly robust software platform that represents over 100 person-years of expertise in solving demanding machine vision and image analysis problems.

Sherlock 7.3.0.0

Sherlock 7.2.7.7 was migrated from VisualStudio 2005 (VS2005) to VisualStudio 2019 (VS2019) and the version was changed to Sherlock 7.3.0.0.

Table of Contents

Preface 3
1. Sherlock System Setup 8
Windows Configuration 8
Minimum System Requirements 8
Software Installation Overview 9
General Recommendations 10
Recommended Order of Installation 10
Get to Know Sherlock 7 11
Starting Sherlock 11
Sherlock Command Line Options 11
2. User Interface Overview 12
How Sherlock Works 12
On-line Help 12
Definitions 12
Sherlock Application Windows 13
Main Window 13
Image Window 14
Program Window 14
Instructions Window 15
Variables Window 15
Watch Window 15
Monitor Window 15
Find Results 16
Report Window 16
Digital Inputs Window 16
Digital Outputs Window 16
Global Image Window 16
SPC Monitor Window 16
Sherlock Menu Bar 16
Program Menu 17
Edit Menu 17
View Menu 17
Run Menu 17
Image Window Menu 17
Options Menu 18
Window Menu 18
Help Menu 18
Sherlock Toolbars 18
Main Toolbar 18
Image Window Toolbar 19
Dialog Boxes 19

Investigation 19
Program 20
Subroutines 20
Flow Instructions 21
The Image Window 21
ROI 21
Readings 22
Viewing an Array of Readings 22
Variables 22
Instructions 22
3. Sherlock Tutorial 23
Tutorial Overview 23
Simple Investigation Loop 23
Start Sherlock 23
Create An Image Window 24
Load an Image 24
Change the Views 24
Create a ROI 25
ROI Options 26
Create a Formula 28
Create Variables 29
Assign the Variables 29
Run and Save the Basic Investigation 30
Modified Program Loop 31
Create Subroutines 31
Conditional Statement 31
Add Output 33
Run the Investigation 35
Save the Completed Investigation 35
4. Options and Details 36
ROI 36
Extracted ROIs 36
Masked ROIs 37
Line ROIs 39
Composite ROIs 39
ROI Options 40
ROI - Processing Tab 40
ROI - Position Tab 42
ROI - Properties Tab 43
ROI - Tolerance Tab 46
Image Window Options 47
Image Window - Source Tab 48
Image Window - Image Logging Tab 50
Image Window - Calibration Tab 52

Image Window - Alignment Tab 59
Image Window - Display Tab 60
Application Options 61
Acquisition Options 65
Input/Output Options 67
Digital Inputs and Outputs 68
Serial Connection 69
TCP/IP Connection 70
Modbus Connection 71
Control Logix Connection 72
GE Fanuc Connection 74
Program Switch Configuration 75
HMI Connection 77
Profinet Connection 78
JavaScripts 79
Online JavaScript Reference 80
Entering JavaScript 80
Concurrent Script Options 81
Monitor Options 83
Conditional Expression 83
Output Reading 84
Color Classifier Trainer 85
5. Camera Calibration 88
How Camera Calibration Works 88
Setting Up a Camera Calibration 88
Program Calibration Using Points 89
Interactive Calibration Using Transform 91
Interactive Calibration Using Grid 93
Multiple Calibrations Per Camera 94
Automatic Recalibration 94
Perspective Distortion 95
6. Alignment 96
How Alignment Works 96
One Point Landmarking 96
Two Point Landmarking 96
Three Point Landmarking 96
Setting Up Part Alignment 97
Landmarking with Search 97
Retraining Alignment 97
Disabling Alignment 97
7. Search Algorithms 98
Pattern Finding Tools 98
Correlation Search 98
Geometric Search 99

Line Based Search 99

Training A Pattern 99

Choosing Between Correlation and Geometric Search or Line Search........ 100

Image Defect and Variation 102

8. Inspect Algorithm........ 107

How Inspect Works 107

Inspect Parameters 107

Training Images........ 108

Don't Care Area Editing........ 108

A. Accelerator "short Cut" Keys 109

B. Annotation Colors 110

Glossary........ 111

Calibration Checkerboard 113

1. Sherlock System Setup

Refer to the ***Sherlock7 Installation Guide*** for specific steps on installing the Sherlock software and supporting acquisition drivers. System configuration and requirements are covered here.

Windows Configuration

If you customize Windows, you may have some difficulty with Sherlock. Keep the following requirements in mind:

1. Sherlock is designed for a display resolution of 1024 by 768 or larger.
2. Screen savers are a very bad idea in machine vision. If the system is continuously inspecting, but unattended, a screen saver will interrupt inspection and greatly slow processing speed.
3. If you are saving images to disk, turning off the hard drives (power save) will cause you to lose some images.
4. Depending on your Video Card, some video modes may be much faster than others. If display speed is critical to your application, experiment with the color depth setting. Test the 8-bit, 16-bit, 24-bit and 32-bit color modes, to find out which is the fastest in your system. The available color settings depend on the windows operating system and video card. The Sherlock Monitor window displays instruction time. The Status bar displays total inspection time. Color operations will not operate correctly with 8-bit color display.
5. In the Performance Options, select "Adjust for best performance" and then enable two options: "Smooth edges of screen fonts" and "Use visual styles on windows and buttons".
6. In the Display Settings or Select Theme, select "Windows Classic" for Windows XP, or select "Windows 7 Basic" for Windows 7.
7. Turn the Windows Firewall off to use GigE cameras, or to communicate with other devices on the network, such as PLCs and HMIs. Or configure your firewall to allow all processes related to the Network Imaging Driver and camera acquisition.

The default "Windows XP" theme with the rounded corners on windows, was shown to use 8% to 12% of the system's processing power. Following steps 5 and 6 can give this boot to your machine vision application.

Minimum System Requirements

The Minimum system requirements for Sherlock 7 are:

- Windows 7 or Windows 10 operating system - all 32-bit and 64-bit editions.
- Windows 7 Embedded Standard (WES7) 32 and 64 bit, as installed on the GEVA products. or Windows 10IoT 64 bit as installed on the GEVA products.
- Processor: IA-32 (x86-32) with SSE2 or x86-64 (AMD64, Intel 64), at least 800 MHz.
- 1 GB of main memory.
- 128 MB of free disk space for installation.
- A VGA display capable of displaying 1024 x 768 pixels with 32-bit color and at least 8 MB of video memory. Note: A 1280 x 1024 display with 32-bit color, 16 MB of video memory and graphics acceleration is recommended.

A faster processor, more memory and disk space, and more capable graphics will all contribute to Sherlock's performance and your enjoyment.

Software Installation Overview

You **must have Administrator privileges** to install and run this software (Sherlock, Sapera, IFC, etc.) because modifications are made to the registry, and some protected directories.

Sherlock may be installed from the Sherlock software download. Extract the contents of the download fie and use Windows explorer to navigate to the installation files. Follow the on-screen directions. A program group will be added to the Start menu list. Installing Sherlock also installs the portions of the Intel Processing Primitives (IPP) runtime library, version 6.2 and the Visual C Redistributables.

Sherlock 7 may be installed under Windows 7 and 10 32-bit or 64-bit operating systems. Refer to the ***Sherlock7 Install & Troubleshooting Guide*** for detailed information on installing software and help troubleshooting common problems.

Sherlock software can be installed alone, for Demo purposes. This allows you to experiment with image processing on image files included with the software, or you can substitute your own image files.

Sherlock software requires a license to acquire images from a camera. There are several options for acquisition software.

- Sherlock with Sapera LT software, and a supported frame grabber: X64-Xcelera series, X64 series, PC2-Comp, PC2-Vision, PC2-Camlink. These frame grabbers support a variety of analog and digital cameras.
- Sherlock with Sapera LT software, and a supported GigE camera: Genie, Genie TS, Spyder 3 GigE.
- Sherlock with Common VisionBlox (CVB) software and a supported third-party GigE camera. You need to purchase a "CVB Acquisition for Sherlock" license from Stemmer Imaging.

The Sherlock license can be stored in a hardware lock (a dongle) attached to your PC or stored on the frame grabber board. Sherlock 7 does not run with a Sherlock 6.x license. Sherlock checks at startup to see if the license is installed. Sherlock runs in "demo mode" if no license is installed. Demo mode supports all product features except image acquisition from the camera or frame grabber. Contact Teledyne Imaging to obtain a permanent software license. Use the License Manager supplied with Sherlock to install the Sherlock software license.

Start ➔ Programs ➔ Teledyne DALSA ➔ Sherlock ➔ License Manager

General Recommendations

You do not have to reboot the computer between each software package installation, but there are a few places where a reboot is highly recommended. Rebooting allows the PC a chance to re-evaluate changes to the registry or new environment variables and paths.

If you must copy the install files to another media such as a USB memory, under Windows Vista or Windows 7 always right-click on the setup file and "Run as Administrator". The Sherlock installation file already contains the elements of IPP6 and VC Redistributable needed. You do not need secondary files or directories to install Sherlock alone.

Sherlock and Sentinel can be installed first or last. They are not sensitive to the order of installation. You can uninstall and reinstall Sherlock or Sentinel without affecting the other software programs, except that uninstalling Sherlock will delete your Sherlock configuration files. One exception is CVB. CVB 10.02 installs an older version of the Sentinel software. You should install CVB 10.02 (Sherlock 7.1.x) before you install the Sentinel on your Sherlock CD or DVD. CVB 11.0 (Sherlock 7.2.x) 32-bit version installs a newer version of Sentinel.

Some Sapera frame grabbers must be installed before you can install the related hardware driver software. Install Sapera LT, shut down the system, install the hardware, reboot, install the hardware driver, reboot again, and run the firmware update.

Recommended Order of Installation

Not all of the software mentioned here will be necessary. Detailed steps for each software package are given in the ***Sherlock7 Installation Guide*** available on the Sherlock CD or on vision systems.

1. Install Sherlock (select 32 or 64 bit).
2. Install Sentinel, if you are going to use a dongle.
3. Install Sapera LT Runtime if using Sapera supported hardware. Sapera LT supports both 32 or 64 bit environments.
4. After installing Sapera LT, either reboot and install the GigE framework software or shut down and install the Sapera hardware, then reboot and install the hardware driver software.
5. Install the CVB Software (CVB Runtime or Image Studio) then install the CVB Sherlock Driver.
6. Install GenICamSetup if you are using a PL-USB (32 or 64 bit). Separate files for the 32-bit or 64-bit installation.

The frame grabber must be installed for successful Sapera driver installation. The PC2-Vision and PC2-CamLink are supported by Sapera LT.

You can install a 32-bit application under a 64-bit OS, but a 32-bit application can never access the hardware drivers under a 64-bit OS. This means you can install 32-bit Sherlock software on a 64-bit OS to run in Demo mode with image files, but you cannot acquire images from any hardware or cameras installed on the 64-bit system.

You can install both 32-bit and 64-bit versions on a 64-bit PC, but it is not recommended. While the two are designed for minimal interaction, the Start menu shortcuts, and "sticky" paths do become confusing.

Get to Know Sherlock 7

If you are new to Sherlock, it is highly recommended that you follow the **tutorial**. There is a copy of the tutorial in the Sherlock User's Manual, and a copy is installed on your system in HTML Help format. The tutorial guides you through many of Sherlock's programming features.

Starting Sherlock

Sherlock may be invoked by double-clicking on the Sherlock icon, or started automatically by placing a copy of the Sherlock icon in the Windows Startup group.

Sherlock has several command-line options which may be used to influence its startup behavior. To invoke Sherlock with a command line option, make a copy of the Sherlock icon by holding CTRL while left-clicking, dragging, and dropping the Sherlock icon. This is the standard Windows procedure for duplicating an icon.

Then use File ➔ Properties to edit the command line invoked by the icon. You may wish to edit the caption of the icon to reflect the configuration you set up.

Sherlock Command Line Options

You can invoke Sherlock from a command line with the following syntax. Note that the flags are all optional, case-insensitive, and may occur in any sequence. If you specify an automatic load file, it must come last on the command line.

You can specify the investigation file name (load file) with no flag, if it is the only command line parameter. If there are any other flags or arguments, you must use the load flag, and the load file must come last.

```
IpeStudio.exe -show:[min|max] -ns -run:[once|cont] -acq:<filename> -load:<filename>
```

-show: min - minimize the Sherlock window on open (and no splash screen)
max - maximize the Sherlock window on open.

-ns - No splash screen. Eliminates the splash screen on Sherlock initialization. This is the default for VisualBasic programs using Sherlock as an image processing engine.

-run: - Run continuously after opening.

-acq: <*initialization file*> - Start Sherlock using the settings in the specified initialization file. Sherlock uses settings in file “Drivers.ini” if no other file is specified. You can supply a different name for the Sherlock initialization file. Alternate names ***cannot*** be specified for the IFC and Sapera LT initialization files (IFCDRV.INI and SAPERALTDRV.INI) in this command.

-load: <*investigation file* > - Load this file after Sherlock initializes. Sherlock looks for the file in the following sequence:

a) Look in the current directory.

b) If the file name has no extension, add .IVS and look in the current directory.

c) Look in the Programs directory under the default installation directory:
(\Program Files\ipd\Sherlock\Programs).

d) If the name has no extension, add .IVS, check the Programs directory under the installation directory.

Example: to start Sherlock, and automatically load a Sherlock investigation “test1.ivs” located in the default directory: **IpeStudio.exe -load:test1** *or* **IpeStudio.exe test1.ivs**

Example: to start Sherlock and run continuously using the same investigation file:
IpeStudio.exe -run:cont -load:test1

2. User Interface Overview

This section describes the Sherlock User Interface. Designed for ease of use, this interface consists of windows, pull-down menus, toolbars, on-line help, and dialog boxes. All these entities follow standard Windows look-and-feel, point-and-click interface conventions. More details are available in "Options and Details" (beginning on page 36).

The best way to become familiar with the function and features of Sherlock, is to read and follow the Tutorial beginning on page 23.

Sherlock provides multiple access methods for many functions. For example, some actions can be accomplished from the Menu bar, from a tool bar, or with an Accelerator Key.

How Sherlock Works

A Sherlock "investigation" program is based on the Image Window and Program Window. You draw one or more ROI ("Regions Of Interest") in the image, and add image processing to the ROI. The measured results are called "Readings". In the Program Window, you add Formulas and Instructions to manipulate the readings, and communicate results or decisions to the user or to peripheral equipment, through the Serial, Ethernet, or Modbus interfaces.

When Sherlock executes an investigation once, each instruction in the main subroutine is executed in sequence. After the last instruction is executed, the investigation terminates. When Sherlock executes an investigation Continuously, each instruction in the main subroutine is executed in sequence, and after the last instruction is executed, the sequence is repeated from the beginning. This repeats indefinitely until the user halts or terminates the investigation.

On-line Help

Sherlock provides on-line help in HTML format. Access the help with the F1 button, or the Help menu. The "active" window determines how help opens (to which page) when you press F1. Sherlock also contains "Tool Tips" and other messages that appear when you hover the cursor over many program elements, and messages in the Status bar. All instructions, preprocessors and algorithms have their own Help topics.

Definitions

Use Sherlock to create and run industrial vision applications, each called an ***investigation***. An investigation is like a computer program, except that you create it in a graphical, point-and-click environment.

Most of the skill in vision engineering is in knowing what combination of lighting, ***preprocessors*** and ***algorithms*** will best differentiate between good and bad. Sherlock frees up vision application programmers' time to concentrate on the processing, instead of programming tasks.

An ***investigation*** is composed of ***subroutines*** which contain instructions Sherlock executes in a user-defined order to perform a particular operation. Sherlock supports the following instruction types: image window, ROI, flow control, variables, and I/O control.

An ***Image Window*** instruction is a virtual camera that snaps a picture for processing. Image Windows appear in their own window within the main Sherlock window.

The ***ROI*** (region of interest) is the basic image analysis Instruction in Sherlock. A ROI is a section of the image contained in the image window. Sherlock displays ROIs as lines and circles drawn within the image window. Use your mouse to draw the different ROI types (rectangle, annulus, torus, area circle, polygon, point, line circle, arc, rake, spoke, bull's-eye, and rainbow) during investigation development. ***ROIs*** perform image processing and analysis on pixels within their boundary, including the pixels on their boundary or outline.

Sherlock returns each analysis result as a ***reading***. Most of the learning curve associated with Sherlock lies in learning what ROIs can do and how to set them up.

A ***formula*** is a subroutine instruction that performs a computation function on input ***readings*** and ***variables***, and produces output ***reading***(s).

A ***reading*** is a single value or array of values, of type number (***N***), point (***P***), line (***L***), Boolean (***B***) or character string (***S***). Sherlock readings are created by ROIs, formulas, or I/O Instructions. Use readings to generate reports, or as inputs to formula instructions. You can apply ***tolerances*** to single readings to make pass/fail decisions and influence execution flow.

A ***variable*** is a location in memory for storage of readings and inputs to formulas. Variables have the same types as readings: a single value or array of values, of type number (***N***), point (***P***), line (***L***), Boolean (***B***) or character string (***S***).

An ***I/O control instruction*** generates, tests, or waits for a particular input or output condition. These instructions may be tied to serial, parallel, analog, file, or reporter. This feature allows Sherlock to make decisions, and to send information to or control external devices.

Sherlock Application Windows

Sherlock uses several windows to display features and allow user control of them. They are the Main Sherlock window, the Image window(s), Program window, Instructions window, Variables window, Report window, Watch window, Monitor window, Digital inputs window, and Digital outputs window.

You can tile, cascade, and minimize Image windows. Image windows are not dockable. All other windows are dockable, can be pinned and unpinned, and can be dropped into other windows to create Tabs in a window. You will see an outline of the attempted dock or drop position. The cursor position determines the docking or drop location, not the title bar of the window. Undocked windows are "sticky" and attempt to dock at all sides of the Main window, as you move them. Hold the Control key to turn off the stickiness as you drag a window.

If you cannot find or see a particular window, pull down the View menu to see if it is checked (active or shown). If the window is checked, it may be a Tab inside another window.

Main Window

The Main window (Figure 1) contains all other windows. The Menu bar and Main Toolbar appear at the top of the Main window. A new installation of Sherlock usually displays one image window and the Program window, inside the Main window. Use the View menu to display the other windows in the Main window.

You can minimize the Main window with the Minimize button at the top left of the window. All displayed windows are minimized. Undocked toolbars are not minimized.

You can minimize the Main window to the Windows System Tray from the Window menu in Sherlock's menu bar (Window ➔ Minimize to Tray). All displayed windows and all undocked toolbars are minimized.

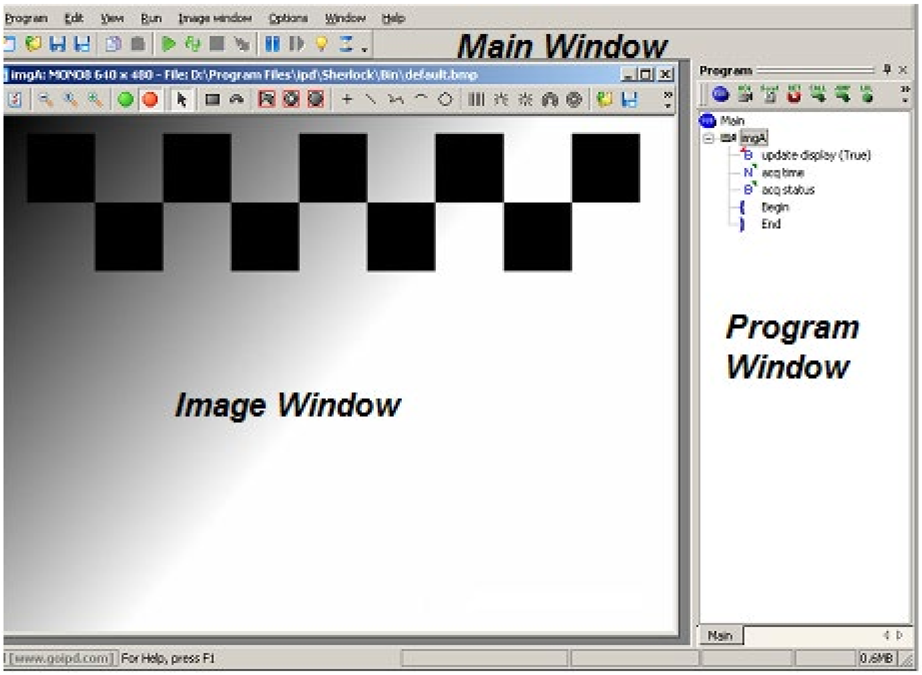


Figure 1. Default Windows

Image Window

The image window (Figure 1) contains and displays the image. You draw one or more ROI (region of interest) in the image window, and add image processing functions in the ROI Properties edit window. You can have more than one Image window in Sherlock. You can tile, cascade, and minimize Image windows. Image windows are not dockable.

The Image Window menu (menu bar in the Main window) controls the active (highlighted) image window. Each Image window can have its own toolbar, or you can turn off toolbars in image windows (Image Window menu) or undock the toolbar and close the undocked toolbar, to save display space.

Create a new Image window in the Program window, either by clicking on the “Create Image Window instruction” button (”ACQ”), or by dragging the “Image window” instruction from the Instructions window into the Program window.

Select the image source for a window by a) clicking on the Options button in the image toolbar, or b) use the Image Window menu and select Options, or c) double-left-click in the image window (outside an ROI).

Use the image window Toolbar, or the Image Window menu, to draw ROIs (regions of interest) in the image. Sherlock performs image processing on the image area inside an ROI.

Program Window

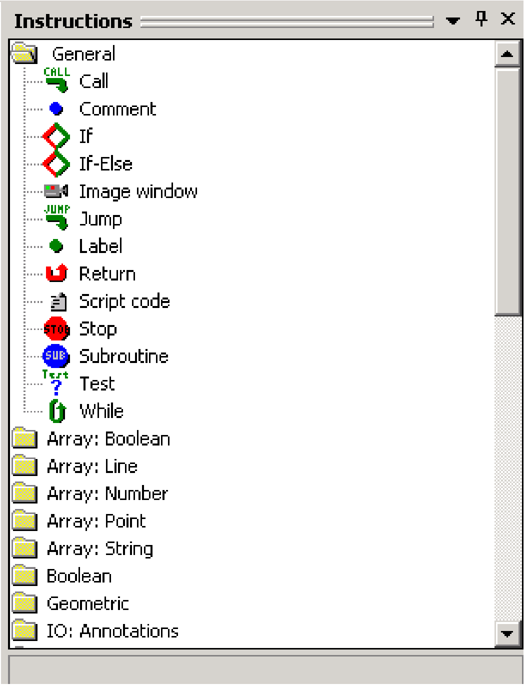
The Program window (Figure 1) displays routines, subroutines, and icons that make up a Sherlock Investigation. (This was called the strategy in Sherlock 6. The Program toolbar supplies buttons for inserting some Instructions (Subroutine, Image window, Flow Control instructions, and JavaScript code).

Right-click on any instruction (expandable items, beside a “+” or “-”) to display the “toggle breakpoint” and instructions menu pop-up. When you run Sherlock, the inspection will pause at each breakpoint. This debugging tool was inherited from Sherlock 6. The various Run modes in Sherlock 7 provide additional debugging aids. The right-click menu changes with context. You can also disable executing an instruction temporarily.

You can pin, unpin, dock and undock the Program window. Unpinned windows create Tabs at one side of the Main window. You can also run the investigation with the Program window closed.

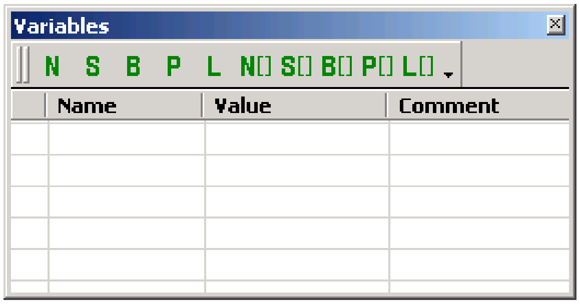
Instructions Window

The Instruction window displays all the available instructions, organized in folders or "toolboxes (The instructions were called Formulas and I/O Instructions, in Sherlock 6.) You can drag and drop instructions into the Program window. You can right-click in the Instruction toolbar to display a list of "toolboxes" and click on one to scroll the Instructions window. You can pin, unpin, dock and undock the Instructions window. Unpinned windows create Tabs at one side of the Main window. You can close the Instructions window to save display space.



Variables Window

The Variables window is used to create and display variables. After variables are created, you can drag a variable into the Program window. Variables can only be created inside the Variables window. You can add a comment in the Variable window by activating the field (F2) or using the variable edit dialog.



Sherlock 6 had several ways to create variables. In Sherlock 7, you can only create variables in the Variable window. You can assign created variables in any place where you previously were able to create them, and you can drag and drop variables. Refer to the Tutorial in the next section.

You can pin, unpin, dock and undock the Variables window. Unpinned windows create Tabs at one side of the Main window. You can close the Variables window to save display space.

Watch Window

The Watch window displays Readings. You drag readings (outputs) into the Watch window, from the Program window, to display their values. You cannot drag variables from the Variables window, but you can drag a reading that is assigned to a variable, from the Program window.

You can pin, unpin, dock and undock the Watch window. Unpinned windows create Tabs at one side of the Main window.

Monitor Window

The Monitor Window displays Benchmark timings and Sherlock internal messages. You can pin, unpin, dock and undock the Monitor window. Unpinned windows create Tabs at one side of the Main window. You can close the Monitor window to save display space.

Find Results

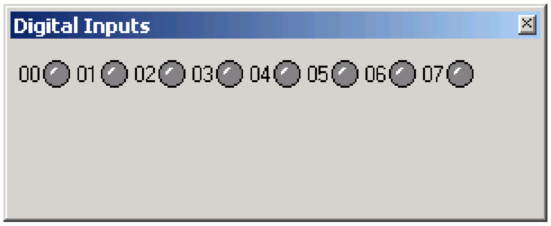
The Find Results window displays the results of Find Name: searching for objects by name. You can double click on an item to jump to its location.

Report Window

The Report window generates reports and collects data for summary reporting. Collected data is sent to the reporter at the conclusion of each inspection. The reporter maintains a 10000 character buffer in which it keeps the last 10000 characters worth of report information generated. You can pin, unpin, dock and undock the Report window. Unpinned windows create Tabs at one side of the Main window. You can close the Report toolbar to save display space.

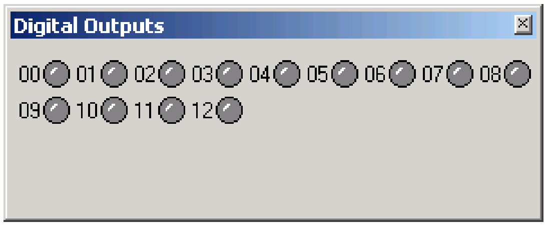
Digital Inputs Window

The Digital Inputs window displays the state (On, Off) of the defined digital inputs Use the IO Instructions (Instructions window) to read the digital inputs. You can pin, unpin, dock and undock the Digital Inputs window. Unpinned windows create Tabs at one side of the Main window. You can close the Digital Inputs window to save display space.



Digital Outputs Window

The Digital Outputs window displays the state (On, Off) of the defined digital outputs. Use the IO Instructions (Instructions window) to read and write the states of the digital outputs. You can right-click on an output to turn it on or off. You can pin, unpin, dock and undock the Digital Outputs window. Unpinned windows create Tabs at one side of the Main window. You can close the Digital Outputs window to save display space.



Global Image Window

This window display the 10 buffers used in some preprocessors. Use Preprocessor "ImgSave" to save an ROI to a buffer (called "bitmap" in Sherlock 6). There are 10 buffers (0 to 9) available for MONO8 images and ROIs. These are temporary buffers which are lost when Sherlock closes. You can change the size of the thumbnails used in the "All" tab using Options➔Application.

SPC Monitor Window

This window displays values for data that you have sent to the SPC, by instructions or by enabling values from the SPC Properties (check icon in the SPC Monitor view

Sherlock Menu Bar

The main Sherlock window contains pull-down menus for access to most Sherlock functionality. Many options are available in several places (menu bar, toolbars, right-click, etc.). The menus and options are also documented in the on-line help.

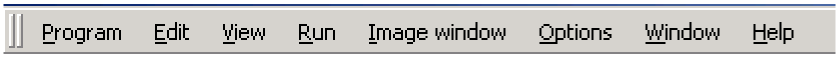


Figure 2. Menu Bar

Program Menu

The Program menu supports the following functions for the investigation file: open a New investigation, Open an existing investigation, Save an investigation file, Save (investigation file) As, Save Description, Open Recent Investigation files, lock and unlock (security) the user interface, and Exit Sherlock

Edit Menu

The Edit menu provides the following functions: Find Name, Copy, Paste selected, Paste Tree , Undo Paste, Delete selected instruction

- Find by name - search for objects by name. *Not all objects can be searched on.* Instructions and variables with names are found. Algorithms, Preprocessors, inputs and outputs do not yield results.
- Paste selected - makes a copy of the selected program item and all of its properties and options.
- Paste tree - makes a copy of the selected item and all attached items below it in the program tree

View Menu

The View menu lists all the available Sherlock windows (except the Image window). Use the View menu to display or hide the Sherlock windows (excluding image windows). A check appears beside the displayed windows. No check appears beside the hidden windows. The default is to display only the Program window. Sherlock will preserve your views and settings when you close and later reopen the application.

Run Menu

The Run menu controls operation or Run Mode of the investigation

- Once - run the Investigation once.
- Continuous - run the Investigation repeatedly. You can stop it by pressing ESC or selecting Halt (below).
- Calibrate - run all subroutines enabled for calibration.
- Halt after inspection completion - stop continuous Investigation, at the end of current cycle. Halt does not disable single step and other debug modes.
- Halt now anywhere - stop after the next instruction. Halt does not disable single step and other debug modes.
- Highlight instructions mode - visually highlights the executing instruction in Run Once or Run Continuous modes.
- Slow execution mode - runs the inspection very slowly, and highlights each step (pressing the shift key speeds execution in this mode).
- Single step execution mode - enables single-step mode. Click on Once to begin the Investigation. Executes one instruction and pauses.
- Execute single step - run the next instruction in single step execution mode or using breakpoints.
- View logged images - opens an image file selection and preview window.

Image Window Menu

The Image Window menu operates on the currently selected or active image window only. The image window menu provides the following operations: Zoom in, out and reset 1:1. Start and stop live image acquisition. Draw an ROI in the image window. Load or Save an image file. Open the Options edit window. Show or hide the Image window toolbar.

Options Menu

The Options menu provides access to Application (Sherlock) preferences, Acquisition hardware (frame grabber and camera) configuration, I/O configuration (digital inputs and outputs, serial, TCP/IP port, and Modbus), and Concurrent javascripts configuration

Window Menu

The window menu operates on multiple Image windows only; not the other Sherlock windows If you have multiple image windows you can tile or cascade open image windows, or arrange icons for minimized image windows. The image windows are listed, and a check appears beside the active image window. The "Minimize to tray" option minimizes the entire Sherlock application, including all undocked toolbars, to the Windows System Tray (beside the system clock).

Help Menu

"Help Topics" opens the Sherlock Help file. "About Sherlock" displays the software version.

Sherlock Toolbars

Sherlock tool bars provide icons or buttons for many features. Icons add point-and-click operation to Sherlock. Sherlock supports tool bars for the Main and Program windows. Sherlock tool bars have tool "tips" for each button. Each tip gives a one-word description of the button function. Activate the tool tip by pausing (hovering) the mouse cursor over a button.

Main Toolbar

This Main toolbar (Figure 3) provides the following functions, listed in sequence:

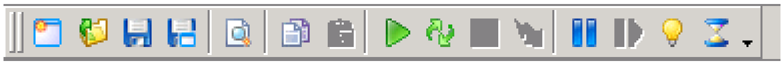


Figure 3. Main Toolbar

- New Investigation - start a new Investigation
- Open Investigation - open an existing investigation
- Save Investigation - save the active investigation
- Save Investigation As - save the active investigation to a new name
- Find Name - search for objects in the investigation
- Copy - copy the selected Instruction(s) to the Windows clipboard
- Paste the instructions in the clipboard.
- Run Once - run the Investigation once then halt.
- Run Continuously - run the Investigation repeatedly until manually halted.
- Stop after completing the investigation - stop at the end of the current investigation loop.
- Stop now anywhere - do not complete the investigation before stopping.
- Single Step mode - enable single-step mode. Use Run Once to begin in this mode. Click this button again to disable single step mode.
- Execute single instruction - perform the next instruction, in single step mode or using breakpoints.
- Highlight executed instructions - highlight each instruction, in Single or Continuous mode. Click this button again to disable highlight mode.
- Slow execution mode - run slowly and highlight each instruction. Click this button again to disable slow execution mode.

Image Window Toolbar

The Image Window toolbar (Figure 4) provides the following Image Window features:

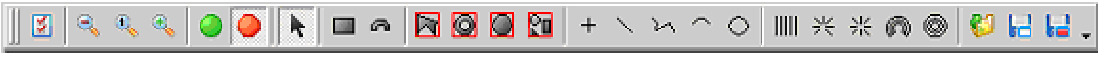


Figure 4. Image Window Toolbar

- Options - opens the Image Window Options.
- Zoom Out - decrease image size.
- Zoom 1:1 - restore image to original size.
- Zoom In - increase image size.
- Live Image - start acquiring image(s).
- Freeze Image - stop image acquisition.
- Select item - changes cursor to normal mode.
- Insert a Rectangle ROI
- Insert a Annulus ROI
- Insert a Polygon area ROI
- Insert a Torus area ROI
- Insert a Circle area ROI
- Insert a User Mask ROI
- Insert a Point ROI
- Insert a Line ROI
- Insert a Polyline multiple-line ROI
- Insert an Arc or curved line ROI
- Insert a Circle line ROI
- Insert a Rake multiple-line ROI
- Insert a Spoke multiple-line ROI
- Insert a Rainbow multiple-line ROI
- Insert a Bull's-Eye multiple-circle-line ROI
- Load a stored image file.
- Save the Image Window image to a user-defined name
- Save the Image Window image with all Graphics to a user-defined name

Dialog Boxes

Sherlock supports many different Edit or Dialog boxes, to edit the properties or attributes of various entities such as Readings, Variables, and so on. The Dialog boxes use standard Windows point-and-click methods. Most Dialog boxes provide help when you press F1.

Investigation

An investigation is a program written in the Sherlock environment. It specifies all details of how an industrial inspection task is to be performed, including synchronization with external devices and report generation. All other Sherlock features exist to support execution of the investigation.

Sherlock stores investigations as files with .IVS extensions. By default, Sherlock stores all investigation files in the \Sherlock\Programs directory, but you can modify this when saving files.

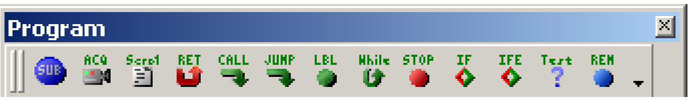
When Sherlock executes an investigation once, each instruction in the main subroutine is executed in sequence. After the last instruction is executed, the investigation terminates. When Sherlock executes an investigation continuously, each instruction in the main subroutine is executed in sequence, and after the last instruction is executed, the sequence is repeated from the beginning. This repeats indefinitely until the user presses halt.

Use the File menu to access investigations. Sherlock keeps track of whether you have modified an investigation. Sherlock prompts you to save an open investigation before loading another one. Sherlock also has an autosave feature that saves the investigation file after a set time period. You can change this time in the Options ➔ Application dialog. The four most recently opened investigations can be quickly re-opened by selecting them from the File menu.

Sherlock supports several run modes. These modes are entered through explicit operator request, or automatically when an investigation file is loaded. Sherlock displays the current investigation mode in the caption bar at the top of the Sherlock window. Use the Run menu to access the different run modes.

Program

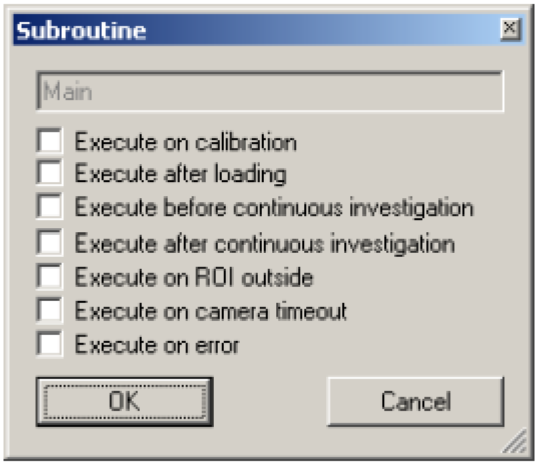
The Program gives structure and direction to an investigation. Sherlock investigations are composed of one or more subroutines. Each subroutine is collection of instructions of type Image Window, ROI, Preprocessors and Algorithms, Instructions, I/O control, Reading, Flow, or Variables. This model is analogous to a conventional programming language. The Instructions window and the buttons in the Program window create these instructions.



Subroutines

There is always at least one subroutine, the "main" subroutine Subroutines allow you to create modular investigations. By putting repetitive groups of instructions into separate subroutines, an investigation can be made smaller, more flexible, and easier to maintain. A non-main subroutine is executed by using a flow instruction's call option. When Sherlock executes a flow instruction that performs a call, the instructions in the called subroutine will be executed. After the last instruction in the called subroutine is executed, or after a return flow instruction is executed, the sequence returns to the original subroutine and executes the instruction after the call instruction.

Sherlock can execute subroutines other times in addition to calling them. In the Program Window, double-click on a subroutine instruction to open the Subroutine Options.



Subroutine

Main

□ Execute on calibration
□ Execute after loading
□ Execute before continuous investigation
□ Execute after continuous investigation
□ Execute on ROI outside
□ Execute on camera timeout
□ Execute on error

Figure 5. Subroutine Options

- During calibration - Execute this subroutine when Investigate ➔ Calibrate (F11) is invoked.
- After loading an investigation file - Execute this subroutine after the investigation is loaded. This is useful for performing one-time initialization, such as turning on I/O or opening a file.
- Before starting continuous investigation - Execute this subroutine immediately after the Continuous mode is selected, but before running the Main routine. This is useful for one-time initialization that is time-critical or should not be left in a "steady state".
- After halting continuous investigation - Execute this subroutine when the Halt button is clicked, immediately after ending the investigation. This is useful for one-time termination that is time-critical or should not be left in a "steady state".
- When the ROI moves outside the Image window - Execute this subroutine when you are tracking movement (Alignment), and the ROI moves outside the image window. This specific error that can be treated separately.
- When a camera timeout occurs - if the camera fails to acquire a new image.

- When an error occurs - any programmatic error other than ROI outside the image window, or a camera timeout.

Flow Instructions

Flow subroutine instructions alter the execution sequence of subroutines and other instructions Flow instructions have three types: Jump, Call, and Return. Use the Flow Edit dialog to add, change, or delete flow instructions.

Jump instructions allow execution to continue at any other instruction in the current subroutine. Be careful not to create an infinite loop! To terminate an investigation that appears to be in an infinite loop, press the ESC key.

Call instructions allow execution to continue at the beginning of any other subroutine. When the called subroutine completes, execution returns to the calling subroutine and resumes with the instruction after the call instruction.

Return instructions terminate execution within the current subroutine. There is an implicit return at the end of each subroutine, so you do not need to add one. Returning from the main subroutine terminates the investigation.

The Image Window

This Image Window is a flexible, virtual camera that acquires a required image. An image window contains data taken from a file or sequence of image files, from another image window, or from a camera attached to GigE network port or a frame grabber input.

Sherlock provides two methods for creating an image window:

1. Clicking the Insert Image Window button in the Program window toolbar.
2. Drag the Image Window instruction from the Instruction window into the Program window.

You can have multiple Image Windows in your investigation.

The image window presents an image for user-directed image processing. This window consists of a title bar, an image area, and a tool bar.

The image window always contains the Image Window Name and the Camera Number or Image File Name in the title bar.

You cannot expand the image window beyond the size of the current camera or file image. Generally speaking, keep the Main window maximized on the screen when working on new Investigations; this maximizes the areas available for Image windows and other Sherlock windows.

The Sherlock Zoom feature allows you to increase or decrease magnification on specified areas of a image window. Use the '=/+' and '-' keys or the magnifying glass icons on the tool bar to control Zoom.

Once you load a stakeout from an image file, it will continue to load from the file until you explicitly tell Sherlock to acquire from a camera (from Image Window➔Options).

You can minimize an image window. The ROIs and processing tools are not affected by minimizing the window. Delete an Image window by clicking on the close button (X) in the upper-right corner of the Image window, or by clicking on the image window instruction in the Program window and pressing the Delete key. Deleting an image window also deletes all instructions attached to it.

ROI

Image windows contain ROIs. ROIs (regions of interest) are portions of the image that provide the pixel data for processing. Sherlock has four types of ROIs: Extracted, Masked, Line and Composite. These four types of ROIs are grouped in an Image Window's toolbar by light, vertical lines.

Readings

Readings store output from Sherlock instructions. You can use readings in formulas and I/O Instructions. You can drag any reading (with or without a variable assignment) into the Watch window. Readings that are assigned to variables will be displayed in the Variables window. In Sherlock 6, tolerances were assigned in the Edit Readings dialog. In Sherlock 7 you create a conditional statement to test the values of readings and variables (as a separate instruction or ROI properties-tolerances tab).

Viewing an Array of Readings

There are two ways to view an array of readings

Hover

1. Run the investigation once, to populate the reading.
2. Hover the cursor over the reading name in the Program window. A pop-up lists the array values.

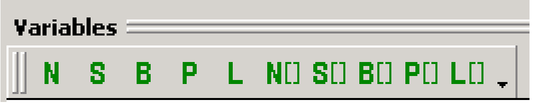
Assign to Array Variable

To display an array of reading values, assign the reading to an array variable.

1. In the Variables window, create an array variable of the correct type.
2. Assign the reading to the array variable. There are two ways to assign a variable:
   a. Drag the Variable into the Program window, and drop it onto the Reading name. Or,
   b. In the Program Window, double-click on the Reading name. This opens the Edit Reading dialog. Use the drop list to assign the reading to the array variable, and click Apply.
3. Run the investigation once, to populate the reading and the variable array.
4. Double-click on the variable name in the Variables window. This opens the Array Variable edit window, that you can scroll through to see the array values.

Variables

Variables can only be created and deleted in the Variables window. The buttons in the Variable window toolbar create single value variables and array variables. Variables can be assigned or unassigned in many places, in dialogs. Variables can be dragged and dropped onto readings. You can right-click on a reading to assign a variable. You can double-left-click on a variable to open a dialog where you can set a value or assign a variable. You can select and delete assignments in the program window. You can only select and delete variables in the Variables window. You will be prompted if you are deleting a variable that is assigned to a reading.



Instructions

The Instructions window does not have a toolbar. You can drag formulas and I/O instructions from the Instructions window into the Program window. You can right-click in the Instructions window to open a list of toolboxes. Selecting a toolbox will scroll the Instructions window to that toolbox.

Array instructions create and manage arrays, insert and extract members in arrays, but do not analyze array values or manipulate numbers. Use Numeric or String instructions to manipulate data or values stored in arrays.

3. Sherlock Tutorial

This tutorial provides a high-level approach to Sherlock features. Subsequent chapters provide more detail, for example about investigation modes, application options, edit dialogs.

Sherlock provides multiple access methods for many functions. For example, you can open a file from the File menu, from the Main tool bar, or with an Accelerator Key combination.

Tutorial Overview

This tutorial shows how to do basic Sherlock image acquisition and processing. It shows all the features of Sherlock, but not all the details of every feature. Later chapters and the on-line help provide the reference material you need to use Sherlock as the powerful, easy-to-use tool it is.

This tutorial assumes that you are familiar with Microsoft Windows. If you are not, run the Windows tutorial.

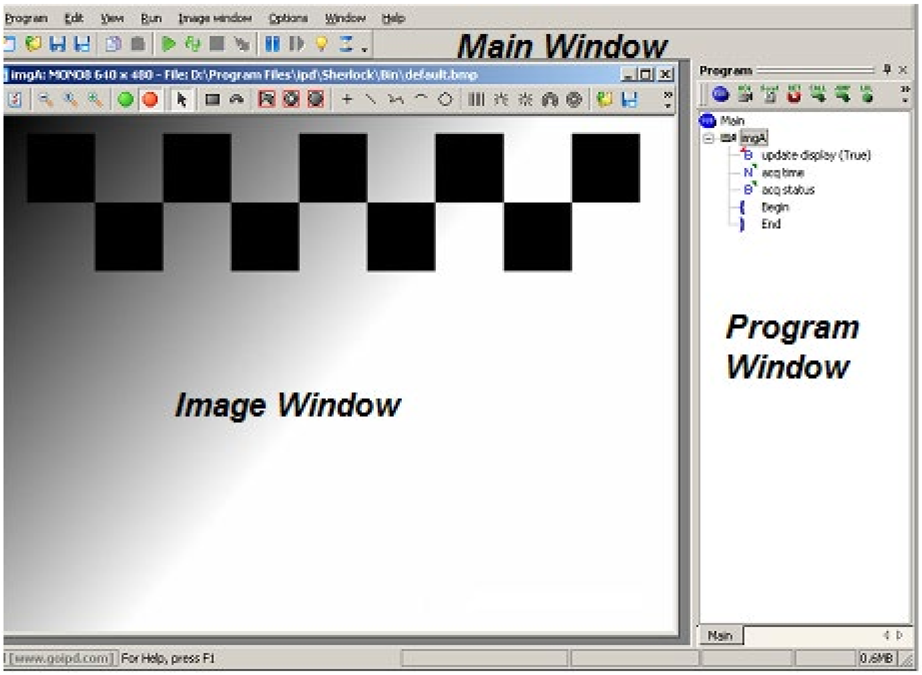
This tutorial can be run in "demo mode" without a Sherlock license, with a Sherlock license using the Camera Simulator, or with a Sherlock license using a frame grabber. There are a few steps that must be skipped if you do not have a Sherlock license. These will be noted.

Simple Investigation Loop

Start Sherlock

1. Start Sherlock by double clicking the Sherlock Icon on your desktop, or in the Sherlock program group.

The Sherlock "splash" screen appears, and then Sherlock displays the image acquisition and processing environment, a main tool bar, the image window, and the Program window (Figure 6). Sherlock defaults to the monochrome image with a checkerboard across the top, unless the defaults have been changed.

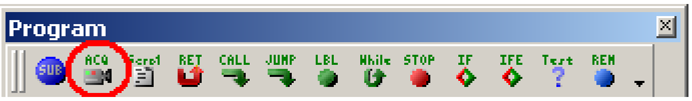


imgA: MONO8 640 x 480 - File: D:\Program Files\ipd\Sherlock\Bin\default.bmp

Figure 6. Default Windows

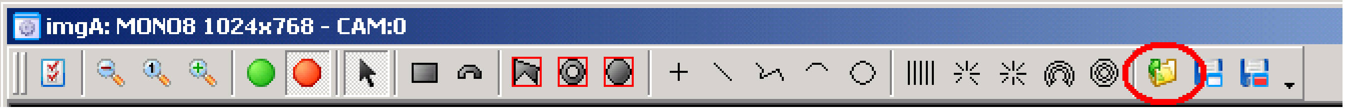
Create An Image Window

Sherlock usually creates an image window in a new investigation, unless this option is disabled. If there already is an Image Window, go to the next step "Load an Image". ***If there is no image window***, click on click the "ACQ" button in the Program window tool bar.



Load an Image

2. Click the Load Image button in the Image window tool bar, and load the image file *MARK1.BMP* from directory Sherlock\images. Figure 7 shows what the image *MARK1.BMP* should look like.



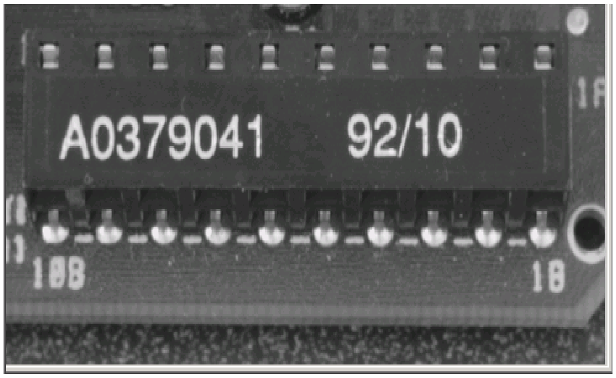


Figure 7. Mark1 Image

Change the Views

3. Go to the Menu bar, and pull down the View menu.
4. Click on Instructions.
5. Pull down the View menu again and click on Variables. Your Sherlock window should now look like Figure 8. The Variables window and the Instruction window are tabs in the lower right pane.

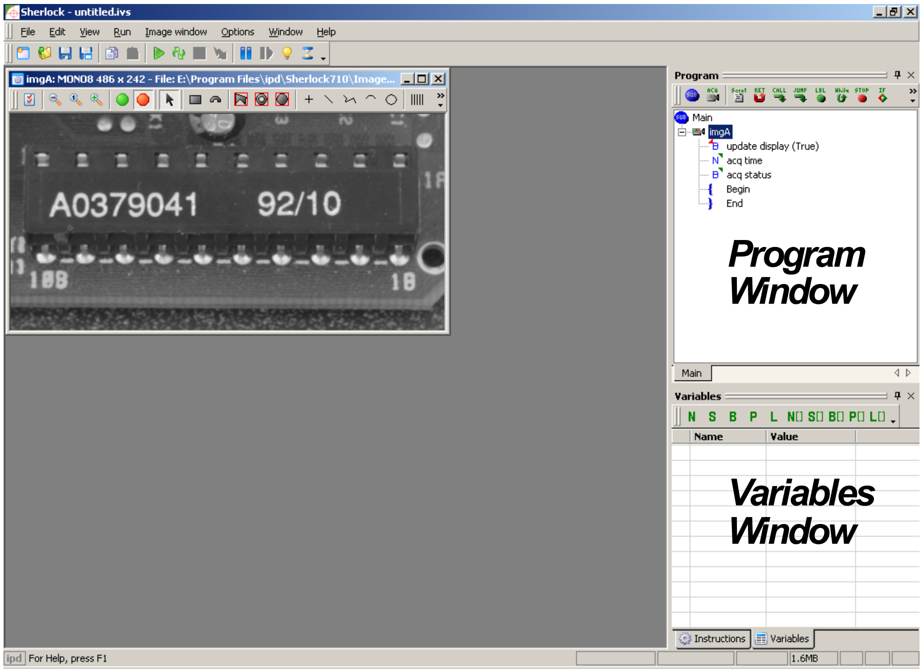
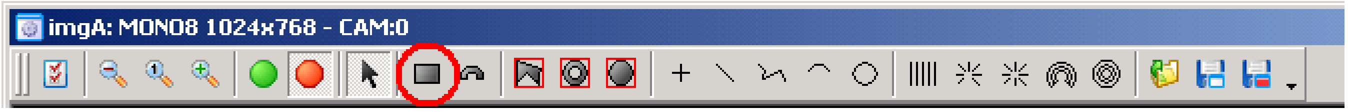


Figure 8. Instructions and Variables

Create a ROI

6. To create a ROI, click on the rectangle button, in the Image Window toolbar.



7. Draw a box around the numbers **92/10** by clicking the left mouse button where you want the upper left corner of the ROI to be, dragging the box to the desired dimensions, and then clicking the left button again where you want the lower right corner to be.

Sherlock creates the rectangle, outlines it with red or green lines (selected or not selected), and places the default name above and to the left of the rectangle ROI. Figure 9 shows what a rectangle ROI looks like.

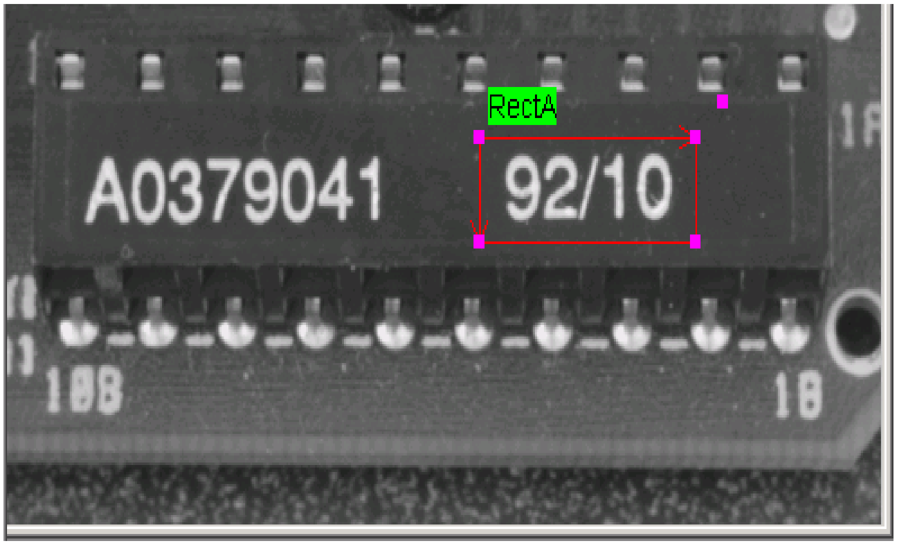


Figure 9. Rectangle ROI

ROI Orientation, Location, and Size

Sherlock supports modifying the location and size of all ROIs, and the orientation (rotation) of rectangle and Rake ROIs. The four corners of the rectangle ROI have Resize handles (solid squares).

To rotate a rectangle ROI, move the mouse cursor to the magenta square above and to the right of the ROI. This is a handle, and Sherlock displays a two-headed arrow when the cursor is over it. Press and hold the left mouse button, then move the mouse to rotate the peek. Release the mouse button to stop rotating the ROI.

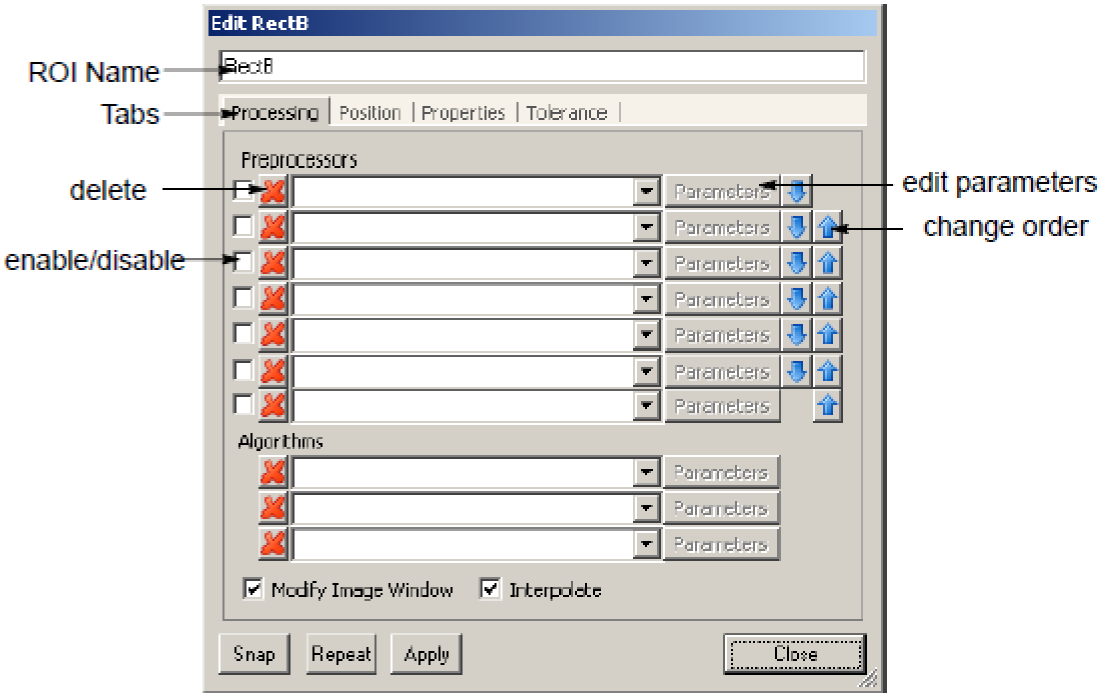
To move the ROI, place the mouse cursor into the center of the ROI. Sherlock displays a four-headed arrow. Hold down the left mouse button and move the ROI to a different portion of the stakeout image.

To resize the ROI, place the mouse cursor on any handle on the bounding lines of the ROI. Sherlock displays a two-headed arrow when the cursor is over this handle. Click and hold the left mouse button, then drag the bounding line to change the ROI size.

ROI Options

8. To display the ROI options dialog: double-click inside the ROI, or double-click on the ROI name in the Program window, or right-click inside the ROI and select “edit”.

The ROI➔Processing options dialog allows you to assign ***preprocessor*** operators, and ***algorithm*** operators. Figure 10 shows the ROI Options dialog.



RectB

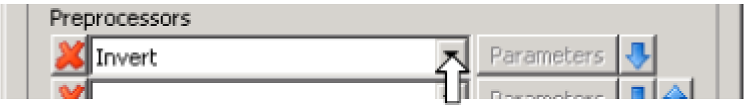
☑ Modify Image Window ☑ Interpolate

Figure 10. ROI Options Dialog

Preprocessor Operators

Sherlock supports up to seven preprocessor operators (Invert, 1X11, 3X3, dilate, and so on) per ROI.

9. To assign preprocessor operators, choose an operator in the preprocessor list boxes (drop list, list box, or combo-box). For this tutorial, select Invert for the first list box.

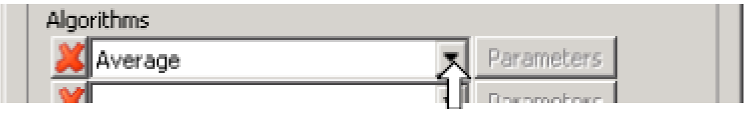


Invert

Algorithm Operators

Sherlock allows you to assign algorithm operators to ROIs.

10. To assign an algorithm, select an operator from the algorithm list boxes below the last preprocessor list box. For this tutorial, select Average.

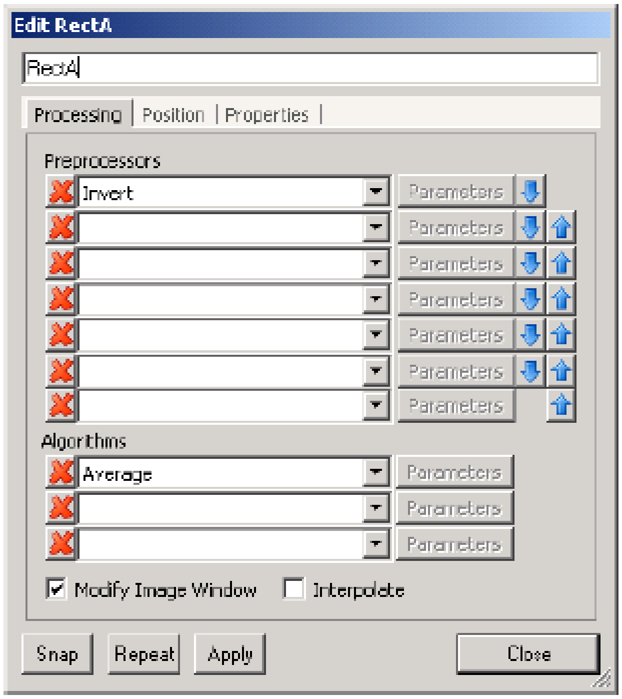


Algorithms

Average

There are other tabs and other options in the ROI Options, that are discussed later in this manual (page 40).

11. Click the "Close" button.



Edit RectA

RectA

Preprocessors

Invert

Algorithms

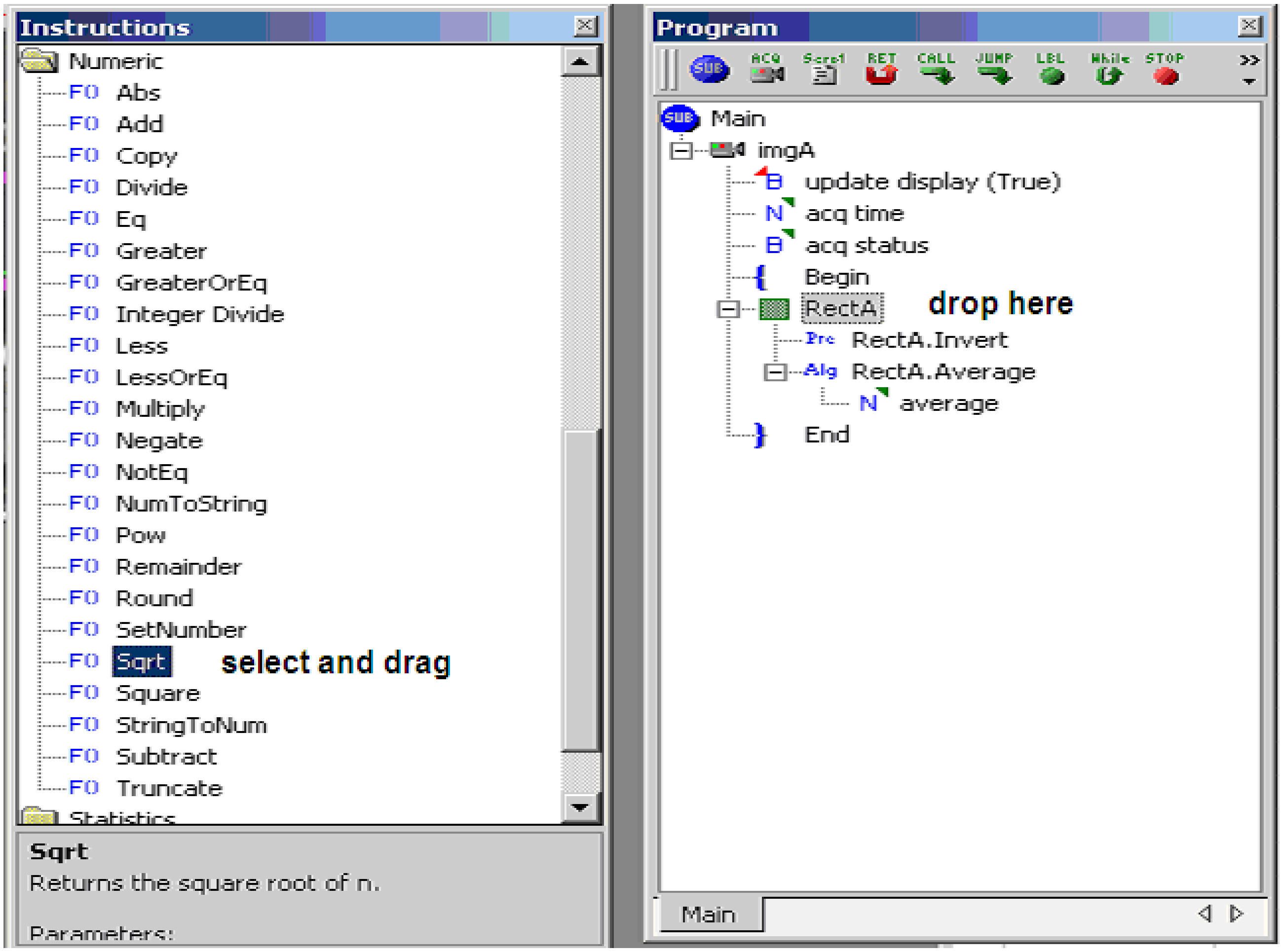
Average

☑ Modify Image Window ☐ Interpolate

Figure 11. ROI Options

Create a Formula

12. In the Instructions window, scroll down to see the "Numeric" folder, and double-click on the folder to open it and list the numeric formulas available.
13. Click on "sqrt" drag it into the Program window, and drop it onto "RectA" (Figure 12).



select and drag

drop here

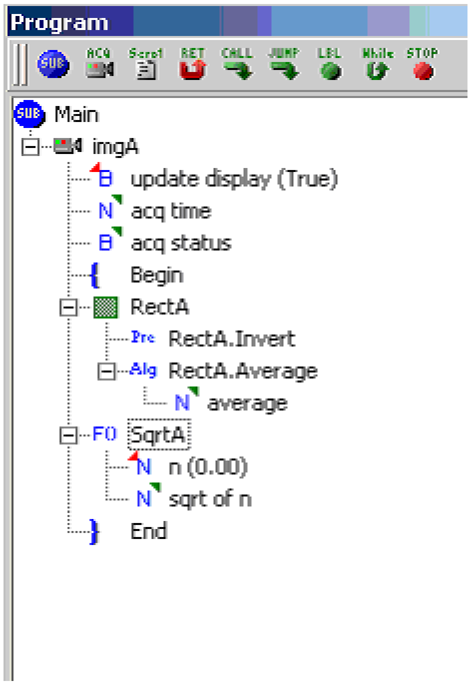
Sqrt
Returns the square root of n.

Parameters:

Figure 12. Drop the Formula onto the ROI

The formula is "created" below the ROI and its image processing (properties), as shown in Figure 13. Below "F() SqrtA" there two lines. The first is "N 0.00" and the second is "N sqrt of n". The "N" specifies these are Number values. The "N 0.00" has a small red arrowhead pointing down and right, at the "N". This arrow symbolizes an input. The "N sqrt of n" has a small green arrowhead pointing up and right, away from the "N". This symbolizes an output reading.

When using a Program window button to create a program item, the new item appears **below** the selected or highlighted program item. When dragging items into the Program window (from another window) the new item appears **below** the item you drop it on. When you drag items within the Program window to rearrange their order, the item appears **above** the item you drop it on.



```
SUB Main
⊟ imgA
    B update display (True)
    N acq time
    B acq status
    { Begin
    ⊟ RectA
        Pre RectA.Invert
        ⊟ Alg RectA.Average
            N average
    ⊟ F0 SqrtA
        N n (0.00)
        N sqrt of n
    } End
```

Figure 13. Created Formula

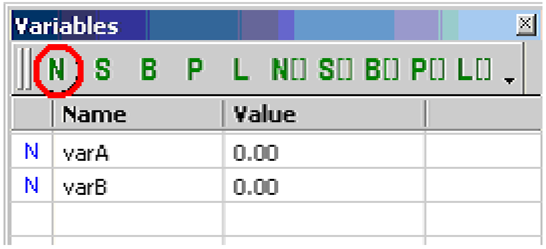
Readings

Readings store output from Sherlock instructions. You can use readings in formulas and I/O Instructions. You can drag any reading (with or without a variable assignment) into the Watch window. Readings that are assigned to variables will be displayed in the Variables window. In Sherlock 6, tolerances were assigned in the Edit Readings dialog. In Sherlock 7 you create a conditional statement to test the values of readings and variables.

Create Variables

You can only create variables in the Variables window (Sherlock 6 allowed creating variables in many places).

14. Click on the "Variables" tab at the bottom of the Instructions window.
15. Create two number variables, by clicking the "N" button twice, in the Variables tool bar.



	Name	Value
N	varA	0.00
N	varB	0.00

Figure 14. Variable Window

Assign the Variables

There are a few different ways to assign variables.

16. In the Variable window, click on "varA" drag it into the Program window, and drop it onto "N average" under "RectA.Average".

17. In the Program window, right-click on "N sqrt of n" and select "connect variable," "varB".

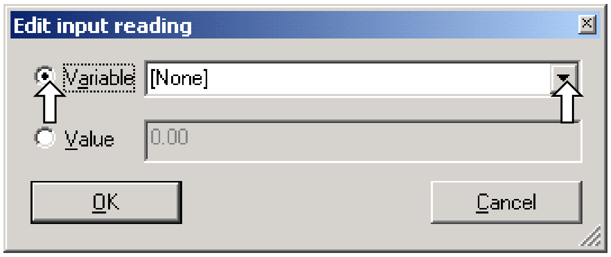
18. In the Program window, double-click on "N 0.00" to open a Edit Input Reading dialog.

19. Click on the radio button beside "Variable" and use the drop list to select "varA".

Notice that the drop list also lets you assign other variables and readings to the input.

The Value field lets you assign a number value.

20. Click "OK" to close the dialog.

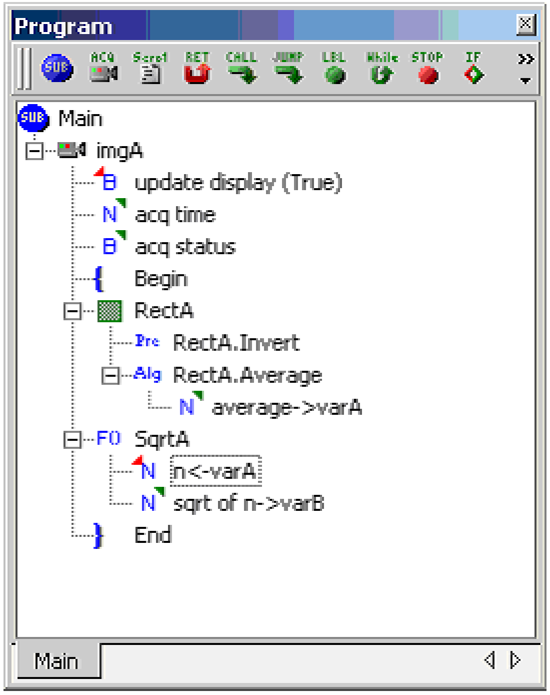


● Variable [None]

○ Value 0.00

Figure 15. Edit Input Reading

The Program window should look like Figure 16. The variables A and B are assigned to the output readings created by the Average and Square root. And variable A is also assigned to the input reading of the Square root. In the Variable window, "varA" and "varB" are still 0.00. You must run the investigation for them to get values.



```
Main
  imgA
      B  update display (True)
      N  acq time
      B  acq status
      {  Begin
         RectA
            Pre  RectA.Invert
            Alg  RectA.Average
                    N  average->varA
     F0  SqrtA
            N  n<-varA
            N  sqrt of n->varB
      }  End
```

Figure 16. Program Window with Variables

Run and Save the Basic Investigation

21. Press **F9**, or in the Menu bar select Run➔Once.

Sherlock calculates the average brightness inside the rectangle and the square root of that number. The values are displayed in the Variables window. The status bar shows the time taken to run the investigation, approximately 8-19 ms to run once (5-9 ms running continuously).

22. In the Menu bar, select File➔Save to save this tutorial investigation. Sherlock prompts you to name this investigation, and adds the file extension ivs.

23. For this example, type **tut_1** and click "Save".

Sherlock 7 has an "auto-save" feature. The default is to save the edited investigation file every 20 minutes. If there are no edits, the investigation file is not saved. You can change this time interval in Application Options (page 61).

Modified Program Loop

Sherlock reading, formula, subroutine, flow control, and I/O control features (and script code in Sherlock Professional) allow you to complete your image processing application with minimum effort.

1. Create two more number variables, by clicking the “N” button in the Variables window toolbar.
2. Double-click on “varC”. This action opens the Variable edit dialog.
3. Change the name to **Number Passed** and click “OK”.
4. Double-click on “varD”.
5. Change the name to **Number Failed** and click “OK”.

Create Subroutines

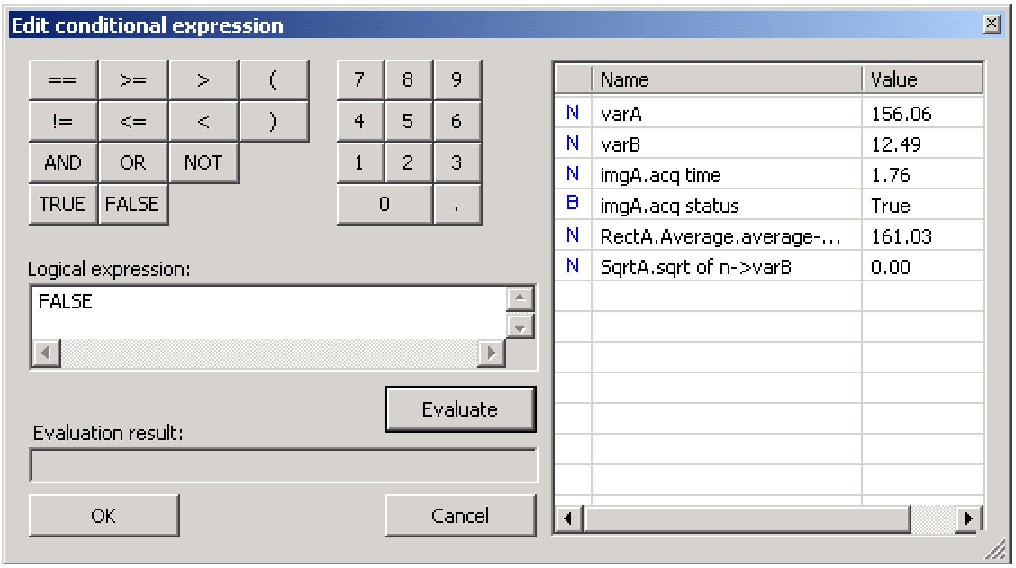
6. Click on the Subroutine button in the Program window toolbar.
7. Click on the Name “subA”. Press F2 (rename) or click a second time, to edit the name. Type **PASS**.
8. Click on the Subroutine button again. Click on the name “subA”. Press F2 or click a second time to edit the name. Type **FAIL**.

At the bottom of the Program window are tabs for Main, PASS, and FAIL.

9. Click on the **Main** tab at the bottom of the Program window. Click on the **Instructions** tab at the bottom of the Variables window.

Conditional Statement

10. In the Program window, click on “F () SqrtA”.
11. Click on the “IFE” button in the Program window toolbar (you may need to use the chevrons >> to see the button), or drag the “If-Else” instruction from the top of the Instruction window and drop it onto the formula “Sqrt” in the Program window.
12. Double-click on the “ifelseA” instruction, to open the Edit Conditional Statement dialog (Figure 17).
13. Select and Delete the statement “FALSE”.
14. Click on the left parenthesis button.
15. Select “varA” and drag it into the Logical expression field.
16. Click on the “>=” button (greater or equal to).



Logical expression: FALSE

Evaluation result:

	Name	Value
N	varA	156.06
N	varB	12.49
N	imgA.acq time	1.76
B	imgA.acq status	True
N	RectA.Average.average-...	161.03
N	SqrtA.sqrt of n->varB	0.00

Figure 17. Conditional Statement

17. Type the number **50**.
18. Click on the right parenthesis button.
19. Click on the "AND" button.
20. Click on the left parenthesis button.
21. Select "varA" and drag it into the Logical expression field.
22. Click on the "<=" button (less than or equal to).
23. Type the number **170**.
24. Click on the right parenthesis button.

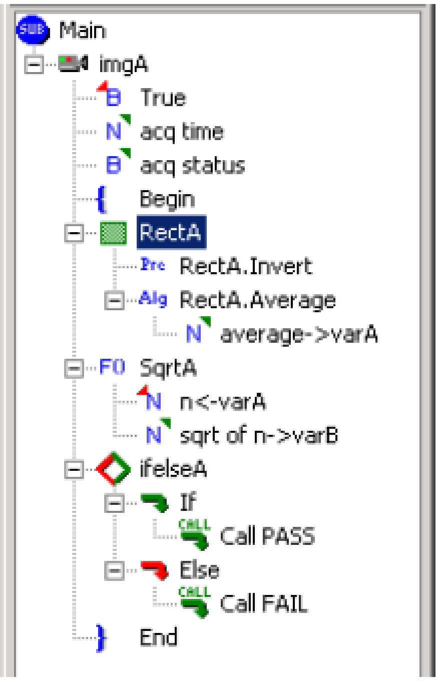
Your Logical Expression should look like this: **([varA] >= 50) AND ([varA] <= 170)**

You can construct complex logical expressions in Sherlock 7. You can test all your readings in one expression, if you desire. The Conditional Statement is also available from the ROI Tolerance tab, allowing you to make your conditional test part of the ROI properties.

25. Click the "Evaluate" button. The Evaluation result should be "TRUE".
26. Click "OK".
27. Click on "If" in the Program window.
28. Click on the "Call" button, or drag the "Call" instruction onto "If".
29. Double-click on "Call" to open the Call to dialog.
30. Use the drop list to select "PASS" and click "OK".
31. Click on "Else" in the Program window.

32.Click on the "Call" button, or drag the "Call" instruction onto "Else".

33. Double-click on "Call" to open the Call to dialog.
34. Use the drop list to select "FAIL" and click "OK".



```
Main
  imgA
    True
    acq time
    acq status
    Begin
    RectA
      RectA.Invert
      RectA.Average
        average->varA
    SqrtA
      n<-varA
      sqrt of n->varB
    ifelseA
      If
        Call PASS
      Else
        Call FAIL
    End
```

Figure 18. Main Routine Completed

Add Output

35. Click on the “PASS” tab at the bottom of the Program window.
36. In the Instruction Window, scroll down to the Numeric folder. (Double-click on the folder to display the contents if they are not visible.
37. Select the instruction “Add” drag it into the Program window, and drop it onto the Subroutine name “Pass”.

NOTE *If you move your cursor over the tabs at the bottom of the Program window, a different Subroutine may open. Drag your cursor over the Pass tab to reopen the Pass subroutine.*

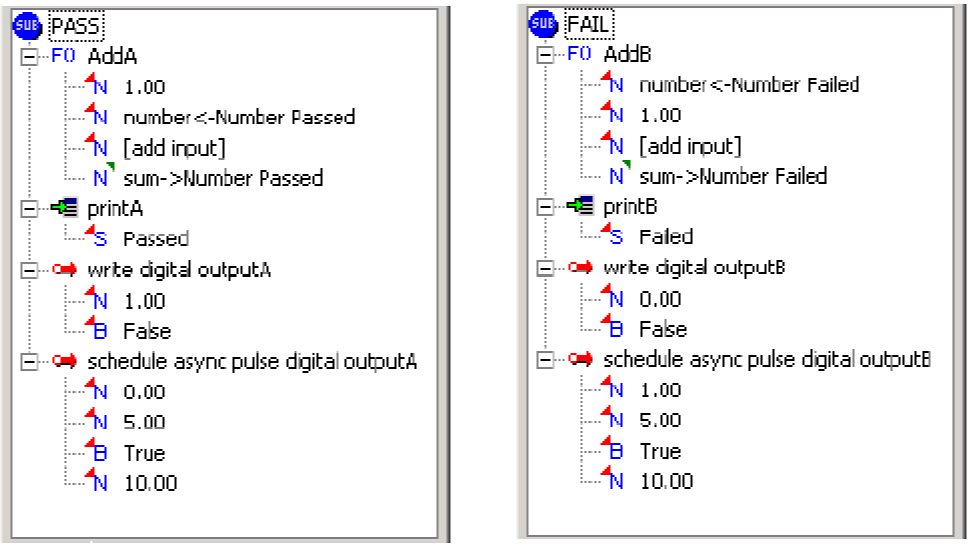
38. Double-click on “N number (0.00)” to open the Edit input readings dialog.
39. In the Value field, change 0.00 to 1. Click “OK”.
40. Right-click on “N [add input]”.
41. In the Connect value pop-up, select variables and then select “Number Passed”.
42. Right-click on “N sum”.
43. In the Connect value pop-up, select variables and then select “Number Passed”.
44. In the Instructions window, scroll up to the “IO:Reporter” folder. Double-click on the folder to display its contents.
45. Select the “Print” instruction, drag it into the Program folder, and drop it on the formula “AddA”.
46. Double-click on “S **Reporter note**”.
47. In the field beside “Value” type **Passed**. Click “OK”.

If you do not have a Sherlock license, and are running in “demo mode” skip forward to step 54. Steps 48 to 53 write to the digital output channels, and will fail if you do not have a license for Sherlock. These steps will execute with the Camera Simulator or with a frame grabber.

48. In the Instruction window, open the IO:Digital Output folder.
49. Select the instruction “Schedule Async Pulse” drag it into the Program window, and drop it onto the “printA” instruction.
50. Double-click on “N channel (-1.00)” and change the value to **0** (for I/O channel 0). Click “OK”. Double-click on “B state (False)” and click the radio button beside “True”. Click “OK”.
51. Double-click on “N time (10.00)”. change 10.00 to 100 or 1000. Click “OK”. This changes the pulse duration.
52. In the Instruction window, select the Digital Output instruction “Write” drag it into the Program window and drop it onto “printA”.
53. Double-click on “N channel (-1.00)” and change the value to **1** (for output channel 1). Click “OK”.

You do not need a “Return” instruction at the end of the Subroutine. There is an implicit Return.

54. Click on the “FAIL” tab at the bottom of the Program window.
55. In the Program window, right-click on the subroutine name “Fail”
56. In the popup menu, move your cursor to “Numeric”. In the submenu, click on “Add”. The “AddB” instruction appears below “Fail”.
57. Double-click on “N number (0.00)” to open the Edit input readings dialog.
58. Type 1 in the Value field. Click “OK”.
59. Right-click on “N [add input]”.
60. In the Connect value pop-up, select variables and then select “Number Failed”.
61. Right-click on “N sum”.
62. In the Connect value pop-up, select variables and then select “Number Failed”.



```
SUB PASS
⊟ F0 AddA
    N 1.00
    N number<-Number Passed
    N [add input]
    N sum->Number Passed
⊟ printA
    S Passed
⊟ write digital outputA
    N 1.00
    B False
⊟ schedule async pulse digital outputA
    N 0.00
    N 5.00
    B True
    N 10.00
```

```
SUB FAIL
⊟ F0 AddB
    N number<-Number Failed
    N 1.00
    N [add input]
    N sum->Number Failed
⊟ printB
    S Failed
⊟ write digital outputB
    N 0.00
    B False
⊟ schedule async pulse digital outputB
    N 1.00
    N 5.00
    B True
    N 10.00
```

Figure 19. Completed Subroutines PASS and FAIL

63. In the Program window, right-click on “AddB”.
64. In the popup menu, move your cursor to “IO:Reporter”. In the submenu, click on “Print”. The “printB” instruction appears below “AddB”.
65. Double-click on “S **Reporter note**”.
66. In the field beside “Value” type **Failed**. Click “OK”.

If you do not have a Sherlock license, and are running in “demo mode” skip forward to the next section “Run the Investigation”. Steps 67 to 71 write to the digital output channels, and will fail if you do not have a license for Sherlock. These steps will execute with the Camera Simulator or with a frame grabber, or a Vision Appliance.

67. In the Program window, right-click on “printB”. In the popup, move your cursor to “IO:Digital Output”. In the submenu, click on “Schedule Async Pulse”. The new instruction “schedule async pulse digital outputB” appears below “printB”.
68. Double-click on “N channel (-1.00)” and change the value to **1** (for I/O channel 1). Click “OK”. Double-click on “B state (False)” and click the radio button beside “True”. Click “OK”.
69. Double-click on “N time (10.00)”. Change 10.00 to 100 or 1000. Click “OK”.
70. In the Program window, right-click on “printB”. In the popup menu, move your cursor to “IO:Digital Output”. In the submenu, click on “Write”. The “write digital outputB” instruction appears directly below “printB” and above “schedule async pulse digital outputB”.
71. Double-click on “N channel (-1.00)” and change the value to **0** (for output channel 0). Click “OK”.

Run the Investigation

1. In the Menu bar, pull down the View menu and click on Reporter.
2. Click on the Green Arrow (run once) in the main window toolbar, or press F9, or use the Run menu to select Run Once.

You should see the value of "Number Passed" in the Variables window change to 1. You should see the word "Passed" in the Reporter window.

NOTE *If the Monitor window opens and displays an error message "Error detected in write digital outputA" the driver for the Frame Grabber or Camera Simulator is not enabled. Save your investigation and then either (1) delete the Digital Output instructions from both subroutines, or (2) correct the driver configuration, (Options ➔ Acquisition) then close and reopen Sherlock, to load the driver.*

3. In the Image Window, drag one corner of the Rectangle ROI, to increase the size.
4. Click on the Green Arrow or press F9 again.

You should see the value of "Number Failed" in the Variables window change to 1. You should see the word "Failed" in the Reporter window.

If you have a Sherlock license you can observe the outputs changing. If you are using a frame grabber and have LEDs connected to Digital Outputs 0 and 1, you would see a 100 or 1000 ms pulse on output 0 when the investigation passes, and a 100 or 1000 ms pulse on output 1 when the investigation fails. You can also open the **Digital Outputs** view, or observe the **Camera Simulator** (if in use). Reduce the size of the Sherlock Main window so you can see the Camera Simulator as the investigation runs.

Save the Completed Investigation

5. In the Image Window, drag one corner of the Rectangle ROI, to decrease the size, to make the program pass again.
6. Click on the Green Arrow or press F9 to run the investigation, and verify passing.
7. Save your investigation to "Tut2.ivs".

4. Options and Details

This section describes in more detail the different options and dialogs found in Sherlock.

ROI

Image windows contain ROIs (regions of interest). ROIs are portions of the image that provide the pixel data for processing Sherlock has four types of ROIs: Extracted, Masked, Line and Composite. These four types of ROIs are grouped in an Image Window's toolbar by light, vertical lines:

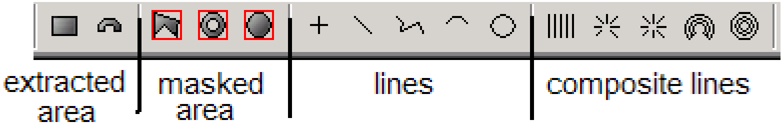


Figure 20. ROI Types

ROIs apply only to preprocessors and algorithms, not to instructions.

Extracted ROIs

The pixels in Extracted ROIs are extracted from the image, into a rectangular buffer. If necessary the ROI pixels are distorted to fit into a rectangle. For example, an Annulus ROI is warped such that pixels towards the outer radius of the annulus are "squeezed together" and those on the inner radius are "stretched out" to form a rectangle of pixels. A rotated rectangle is extracted to a rectangle.

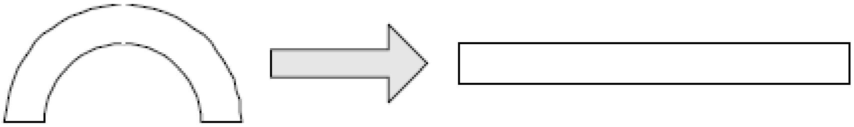
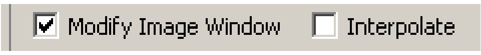


Figure 21. Extraction to Rectangle

You should therefore use caution when making measurements on extracted ROIs. Extracted ROIs can be quickly processed because pixel access is simple and uniform. Line and Composite ROIs are also extracted ROIs, but are one-dimensional.

Interpolation

Either nearest-neighbor interpolation or bilinear interpolation can be used when extracting pixels from an input image The type of interpolation is selected by the Interpolate check box in the ROI's Processing Options.



☑ Modify Image Window ☐ Interpolate

With the Interpolate box unchecked, nearest-neighbor interpolation is used. This means that the pixel closest to the desired pixel position is used. This is really no interpolation.

With the box checked, bilinear interpolation is used to estimate the pixel value at a fractional pixel location. Bilinear interpolation produces more accurate pixel values in extracted images, but is slower than nearest-neighbor interpolation.

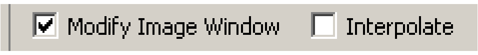
Interpolation is not necessary if you are using non-rotated, rectangular ROIs, and is not necessary (or available) on Masked ROIs. On Line and Composite ROIs, interpolation is linear rather than bilinear.

NOTE *In color images, the "Interpolate" check box does nothing. Interpolation of color images can introduce spurious colors, and is therefore not provided.*

Result Image Injection

Each ROI has a pipeline of preprocessors and algorithms. Preprocessors modify the pixel values and algorithms can also modify pixel values. At the end of the ROI's processing pipeline, the resulting image can be "injected" back into the Image Window for display and additional processing.

If the "Modify Image Window" box is checked in the ROI's Processing Options, then the processed image will be injected into and modify the Image Window.



☑ Modify Image Window ☐ Interpolate

This box is checked by default for Extracted and Masked ROIs, but unchecked for Line and Composite ROIs. Line and Composite ROIs are usually used only for measurements (Algorithms) that don't modify the image.

The ROI's processed image is injected into the Image Window without interpolation - using nearest-neighbor selection of pixel values. This introduces errors in the modified areas of the Image Window. These errors include errors in position and pixel values. For example, create an Annulus ROI, fill it with a constant intensity value and inject it back into the Image Window (Modify Image Window is checked). You will see something like Figure 22.

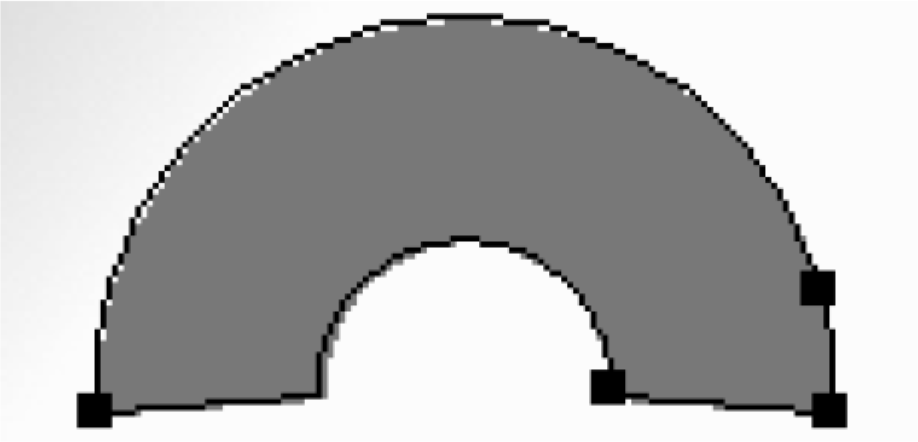


Figure 22. Image Injection Errors

The injected pixels do not all match the area defined by the Annulus ROI. As the injected pixels are all one value, errors in pixel values are not apparent. Use caution when "cascading" processors between ROIs. It is better to use the Image Extract algorithm and Image Inject preprocessor to cascade ROI processing, because these pass the extracted (and interpolated) image rather than going though "un-extraction" and another extraction (and interpolation) in the Image Window.

Masked ROIs

Masked ROIs indicate which image pixels are to be used by a run-length encoded (RLE) description of the ROI and by an optional *mask image*. Masked ROIs have advantages of not distorting distance and area measures within the ROI and do not need interpolation, but can be slower to process than Extracted ROIs because the access to the pixels and boundary conditions are more complex.

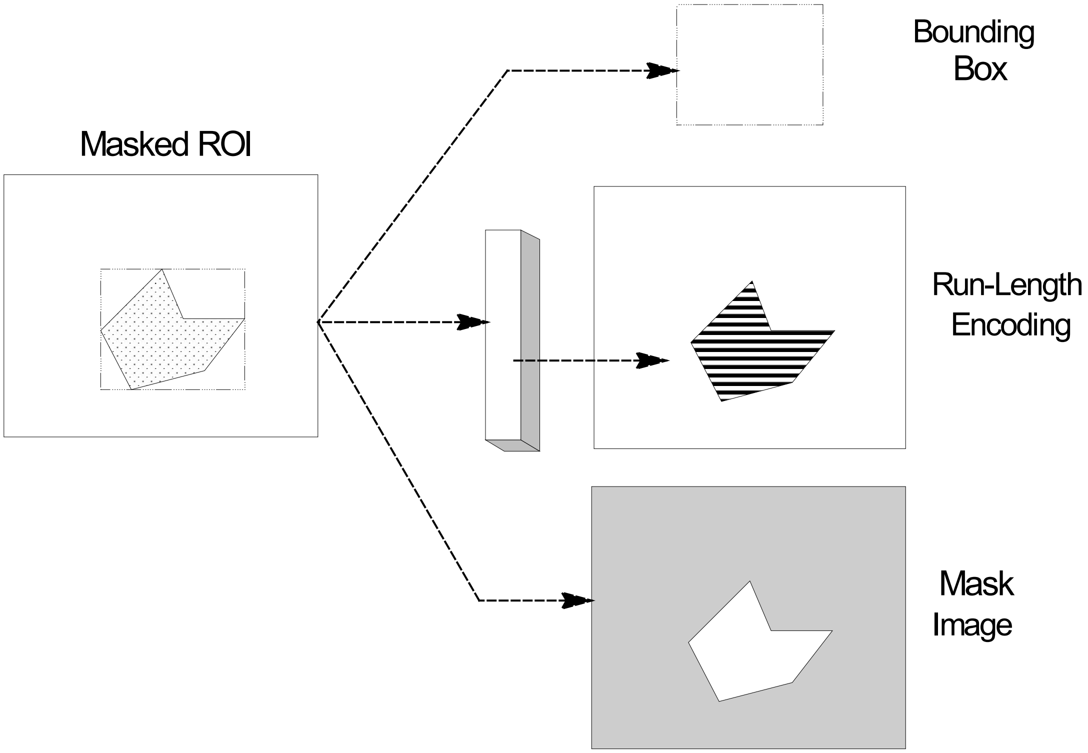


Figure 23. Masked ROI

For example, a polygon ROI drawn in the image window, is shown at the left of Figure 23. Sherlock computes a Bounding Box (a rectangle that just encloses the ROI), a Run-Length Encoded representation, and an optional Mask Image.

RLE Representation

A Masked ROI always has a Run-Length Encoded (RLE) representation This representation has a table of pointers to linked lists, one pointer for each row of the Image Window. Each linked list holds image segments from the ROI that are in that row. Each linked list element consists of a starting X (horizontal) location of an image segment, the length of the segment (in pixels), and a pointer to the next linked list element or NULL if there are no more segments in this row.

To use an RLE representation, Sherlock iterates through the table of pointers to linked lists of image segments, following each linked list to get the segments for that row in the Image Window.

Mask Images

A mask image is created from the RLE Representation This is an 8-bit image where values of 255 indicate pixels that are in the ROI and values of 0 indicate pixels that are not in the ROI. The mask image is the size of the entire Image Window. Sherlock always automatically applies the mask image after a function exits, so only pixels within the Masked ROI area are changed.

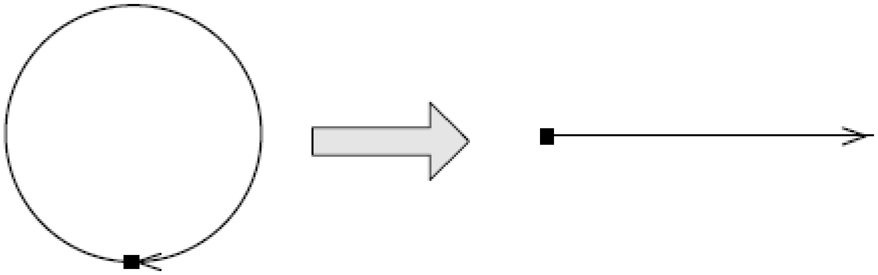
The mask image is useful when processing has boundary conditions. Suppose we want to apply a 3x3 "sharpening" convolution filter to a Masked ROI. At the edges of the ROI, the code to adjust the convolution when not all 9 pixels are in the ROI is complex and slow. Instead, we apply the convolution to the rectangular image region defined by the ROI's bounding box and then let Sherlock automatically select which pixels get changed (in the ROI) in the output image, and which do not (outside of the ROI).

Another advantage of using a mask image is that processing on rectangular areas is generally faster because the pixel access is simpler and because parallel processing instruction sets (such as MMX) can be applied. The major disadvantages of using a masked image are the time to create the image, the storage required, and the time to "post process" your processed image with the image mask. The post processing step (selecting only pixels in the Masked ROI for output into the image) cannot be disabled if you use a mask image.

Line ROIs

Line ROIs (Point, Line, Polyline, Arc, and Circle) are just small, extracted images and are used like Extracted ROIs Line ROIs have an interpolation option, as with Extracted ROIs, but the interpolation is linear rather than bilinear. Color interpolation is not supported.

Point ROIs are treated as 1 x 1 images and are a special type. Line images are put into a 1 x N (row) image. Polyline ROIs are "flattened" into one longer 1 x N image. Arc ROIs are also "flattened" into a 1 x N image. The Circle ROI is broken where the (magenta) square and (red) arrow point meet and unwrapped into a 1 x N image starting at the (magenta) square.



Line ROIs are usually used for measurements (Algorithms) and do not need to modify the Image Window at the end of the ROI's processing chain. Therefore, the default for Line ROIs is to not have the "Modify Image Window" box checked.

□ Modify Image Window □ Interpolate

Algorithms and preprocessors that work on single pixels (points) can often use the same code for Extracted, Line and Composite ROIs. When image areas are needed, as with convolution or morphological operations, then the code must be different. For example, the Erode preprocessor on a Line ROI works along the line (a 1 x n image), while the Erode preprocessor on a Rectangle ROI works on a small image area (m x n).

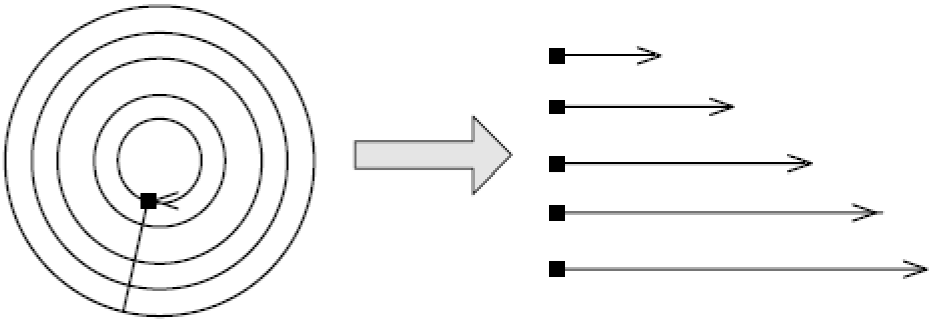
Composite ROIs

Composite ROIs (Rake, Rainbow, Bulls Eye, Spoke and Partial Spoke) are made from multiple line segments, or from multiple Line ROIs. As with Line ROIs, each Composite ROI line is extracted (with or without interpolation) into a set of 1 x N images. Color image interpolation is not supported.

A preprocessor or algorithm is called once for each of the line segments in the Composite ROI, and get the "flattened" and optionally interpolated line segment as 1 x N image.

Composite ROIs are generally used for measurements and generally do modify the Image Window. Therefore, the default for Line ROIs is to have the "Modify image" box not checked.

Many algorithms and some preprocessors can use the same code for Extracted, Line and Composite ROIs. Composite ROIs are also "flattened" into a series of 1 x N images. The "Bullseye" (bull's eye) ROI is broken where the (magenta) square and (red) arrow point meet, and unwrapped into a 1 x N image starting at the (magenta) square. The Rainbow ROI is flattened as well.



ROI Options

To open the ROI Options dialog "Edit ROI" (Figure 24).

- double-left-click on the ROI in the image window,
- double-left-click on the ROI instruction in the Program window,
- right click on the outline of any ROI, or inside an area ROI, and select "edit" in the pop-up.

ROI name - You can rename the ROI by typing a new name in the text field at the top of this dialog. You can also rename the ROI in the Program window, by clicking on the name to highlight it, then press F2 (rename) and type a new name.

Tabs - The Edit ROI dialog has four tabs across the top, for Processing, Position, Properties, and Tolerance. The buttons at the bottom of this window remain, when you click on the tabs.

Snap - update the image inside the ROI, without applying processing.

Repeat - update the ROI image and apply processing repetitively until this button is clicked again. The button changes to "stop".

Apply - apply the processing or property changes, once.

ROI - Processing Tab

Select preprocessors and algorithms for this ROI. Preprocessors are applied sequentially. Changing the order of preprocessors can change the final result.

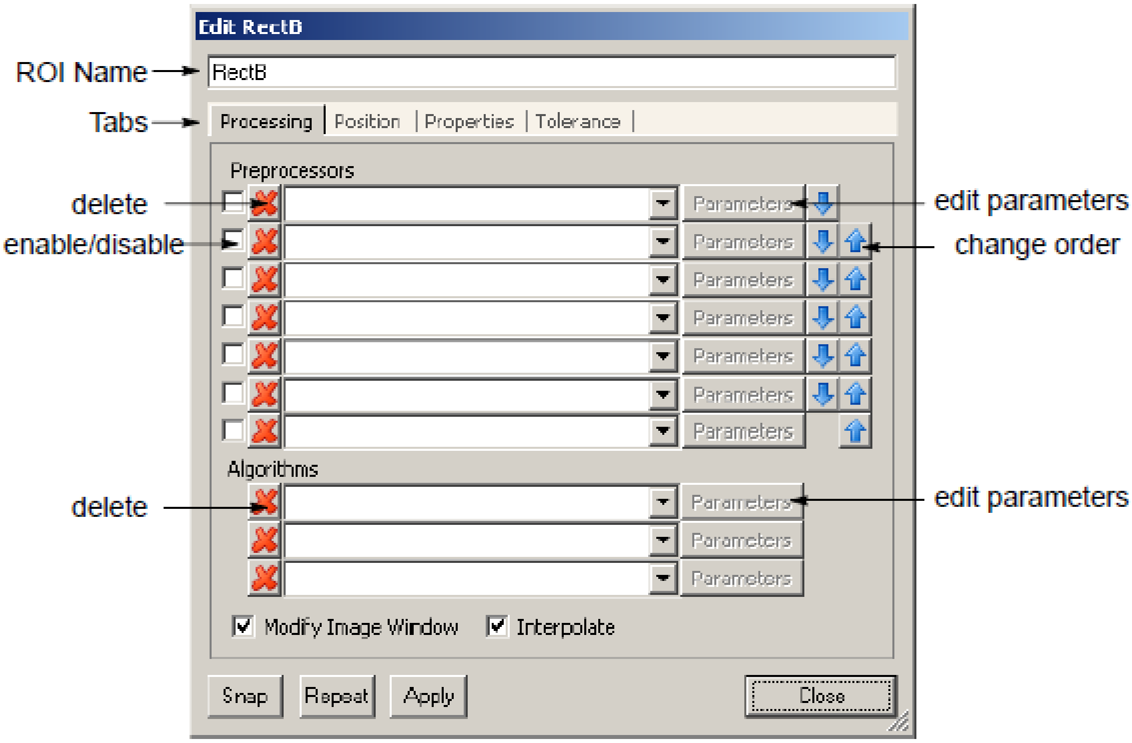


Figure 24. ROI Processing

Preprocessors - use the drop list or combo box to select a preprocessor. The list of available preprocessors changes with the ROI shape and the image type.

- *The checkbox* enables or disables the processor. You can temporarily disable a preprocessor without deleting it. This saves any changes you made to the parameter settings.
- *The X button* deletes the preprocessor.
- *The Parameters button* is active if there are parameters you can change. Not all preprocessors have parameters. Click on the Parameters button to edit. Refer to "Editing Parameters".
- *The blue arrows* move a preprocessor up and down the list. Preprocessors are applied sequentially, in the order they appear. Changing the order can change the results.

Algorithms - use the drop list or combo box to select a algorithm. The list of available algorithms changes with the ROI shape and the image type. Algorithms are applied sequentially, in the order they appear. Algorithms are independent. The order does not change the result, unless variables are used to share values between algorithms.

- *The X button* deletes the algorithm.
- *The Parameters button* is active if there are parameters you can change. Not all algorithms have parameters. Click on the Parameters button to edit. Refer to "Editing Parameters".

Editing Parameters

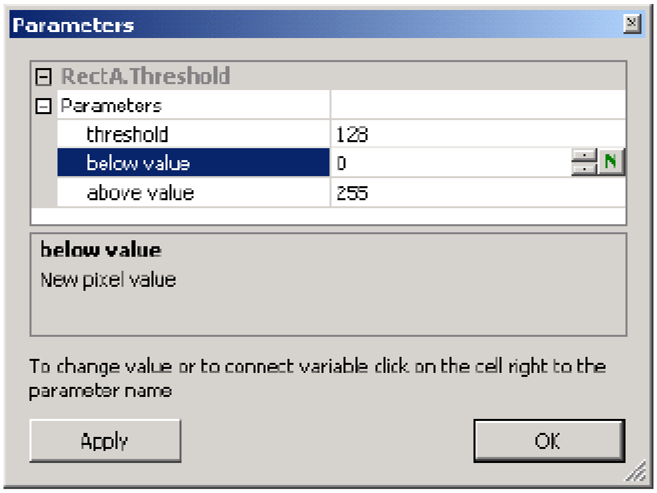
Most of the Edit Parameters dialogs offer a standard interface for editing values. Click in the value field on the right, to activate the field and allow changes. The name of the parameter is highlighted when the value field is active.

Value or Variable: The "**N**" button shows the parameter is a value. Click on the **N** to change the filed to a variable. The button changes to "**var**". The value field becomes a drop list to display all the variables in the investigation.

Numbers - number fields usually have up and down arrows, to allow you to increase or decrease the value by one for each click. You can highlight the number value and type a new value. Number parameters can be assigned to variables, by clicking the “N” button to change it to “var”. The drop list now lists the global variables available.

Booleans - boolean fields have a drop list to assign True or False. You can double-left-click in the field to toggle the value between True and False. Boolean parameters can be assigned to variables, by clicking the “N” button to change it to “var”. The drop list now lists the global variables.

Enumerated values - enumerated fields have a drop list, that lists the defined values. You can double-left-click to step through the defined values. Enumerated parameters can be assigned to variables, by clicking the “N” button to change it to a “var”. The drop list now lists the global variables available.



threshold	128
below value	0
above value	255

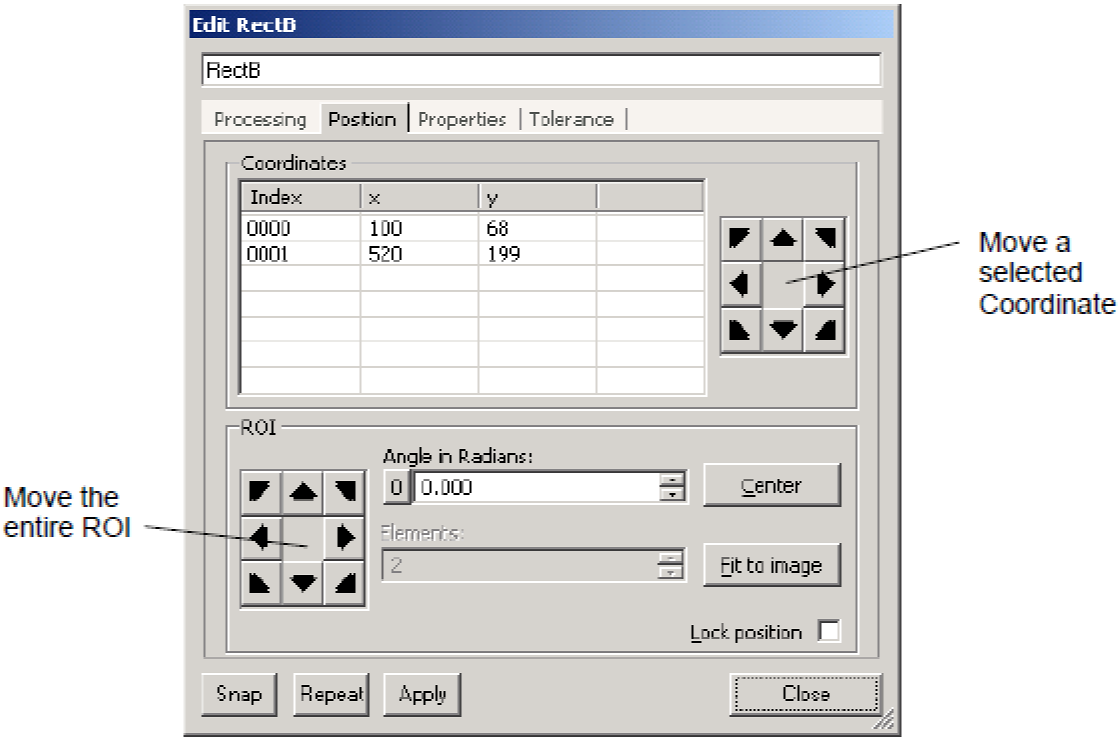
below value
New pixel value

To change value or to connect variable click on the cell right to the parameter name

Figure 25. Edit Parameters Dialog

ROI - Position Tab

You can change the position and rotation of the ROI. The top half of this dialog applies to individual points. The bottom half applies to the whole ROI



RectB

Index	x	y
0000	100	68
0001	520	199

Angle in Radians: 0 0.000

Elements: 2

Lock position □

Move a selected Coordinate

Move the entire ROI

Figure 26. ROI Position

Coordinates - the points where you clicked in the image, to draw the ROI, appear in this list in the order you clicked. Click on a point, and use the *arrows* to the right of this list under "Selected points" to move that coordinate point. This changes the size or shape of the ROI.

ROI - use the *arrows* to move the whole ROI in the image window.

Angle - enter a value, or use the arrows, to rotate the ROI. This applies to Rectangle or Rake only.

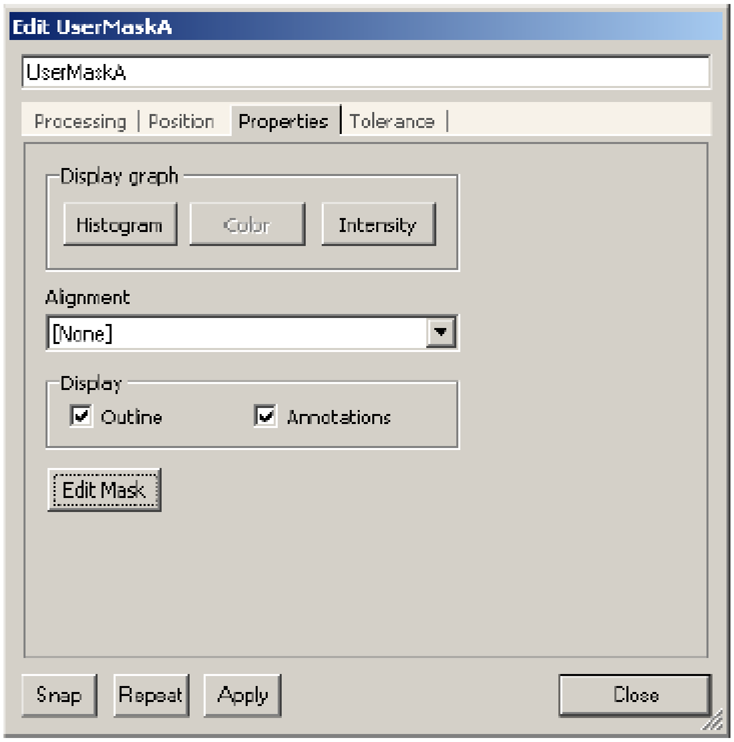
Elements - enter an integer number, or use the arrows, to change the number of lines in a composite ROI. This applies to Rake, Spoke, Rainbow, and Bullseye only.

Center - move the ROI to the center of the image window.

Fit to image - increase the ROI size, to the entire image window. This applies to Rectangle only.

ROI - Properties Tab

Display properties of the ROI, and assign an alignment system.



Edit UserMaskA

UserMaskA

Alignment: [None]

Display: ☑ Outline ☑ Annotations

Figure 27. ROI Properties

Histogram** or **Profile - button changes depending on the ROI. "Histogram" displays a histogram for an area ROI (see page 44). "Profile" displays a line profile for a line ROI (see page 45).

Color - displays a 3-D color distribution map in R-G-B color space (see page 45).

Intensity - displays a 3-D distribution map of the pixel intensities in the ROI X,Y coordinates (see page 45).

Alignment - the drop list shows the available (already created) alignment systems that can be assigned to this ROI.

Outline - show or hide the outline of this ROI in the image window.

Annotations - show or hide the annotations (readings) for this ROI in the image window.

Edit Mask - opens the Mask Editor for a User Mask ROI. *This button is only active for the User Mask ROI.* You cannot resize the bounding rectangle of a User Mask ROI. You draw shapes inside a User Mask ROI. The area inside these shapes is the processed RLE image (see page 38).

Histogram

The Histogram graph/plot shows the intensities from 0 to 255, and the number of pixels for each intensity. A monochrome image shows one plot line. A RGB color image shows three plot lines.

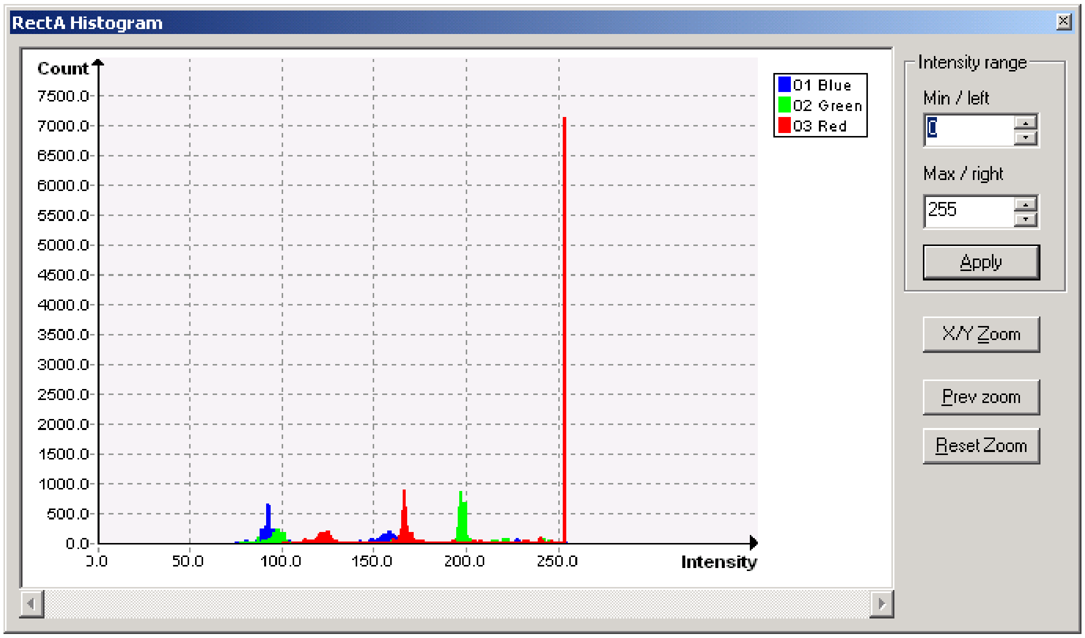


Figure 28. Histogram

- *Min/Left* - You can change the lower range on the Intensity axis, by entering a number or using the arrows, then click "Apply".
- *Max/Right* - You can change the upper range on the Intensity axis, by entering a number or using the arrows, then click "Apply".
- *X/Y Zoom* - Click the "X/Y Zoom" button then draw a rectangle (hold down the left mouse button and drag) in the graph. The area in the rectangle will be expanded or zoomed. You can repeat drawing rectangles to zoom in again. Click "Prev zoom" to go back one zoom step. Click "Reset Zoom" to return to the original display.
- *Measurement Lines* - You can right-click on an axis, then click inside the graph to create a measurement line. You can use the line(s) to get a count or intensity at any specific point in the graph. You can drag a line across the graph to a new location. You can create multiple lines from both axes. Right-click on a measurement line to remove it.
- *Move to Front* - In the histogram for an RGB image, click on a plot line to move it in front of the other plot lines. In the histogram of a mono image, clicking on a plot line highlights the line.

Line Profile

The Line Profile shows the intensity and the gradient (second differential) at each position along the line (monochrome images only). Gradient shows the polarity and rate of change in pixel intensities. This information is useful for edge-detection algorithms.

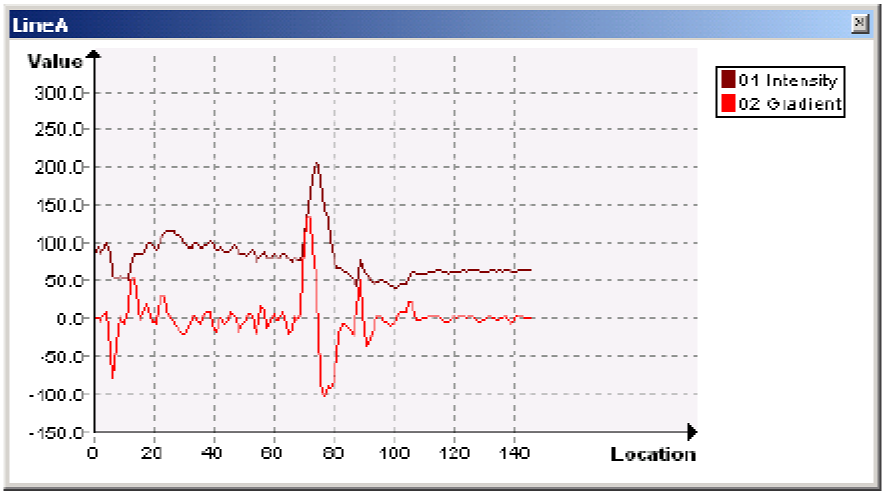


Figure 29. Line Profile

NOTE *You may have to resize this window if Color or Intensity was opened or resized.*

- *Measurement Lines* - You can click on an axis, then drag or click inside the graph to create a measurement line. You can use the line(s) to get a count or intensity at any specific point in the graph. You can create multiple lines from both axes. Right-click on a measurement line to remove it.
- *Highlight* - click on a plot line to highlight the line. Click on the background to un-highlight.

Color Plot

Displays a 3-D color distribution map in R-G-B color space. Click and drag, to rotate the 3-D map. Right-click in this window to open options for this display:

- *Copy to clipboard* - to paste into another program document.
- *Set background color* - select a color for the background of this graph.
- *Data point shape* - sphere or block shape for data points.
- *Data point size* - variable or fixed size data points.

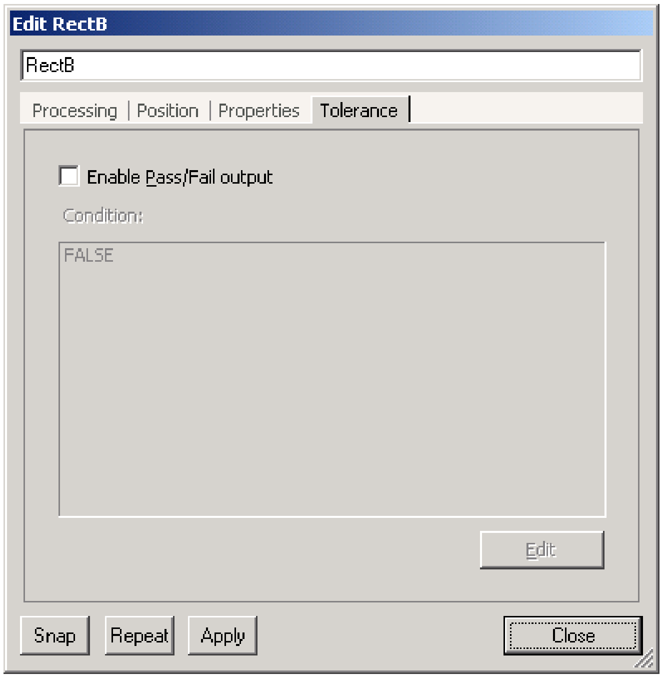
Intensity Plot

The Intensity plot displays a 3-D distribution map of the pixel intensities in the ROI X,Y coordinates. Click and drag, to rotate the 3-D map. Right-click in this window to open options for this display:

- *Copy to clipboard* - to paste into another program document.
- *Set background color* - select a color for the background of this graph.
- *Data point size* - variable or fixed size data points.

ROI - Tolerance Tab

Create a Pass/Fail test condition, attached to the ROI. This is useful when duplicating the ROI for multiple "sensors" in the image area. The condition is copied with the ROI.



Edit RectB

RectB

☐ Enable Pass/Fail output

Condition:

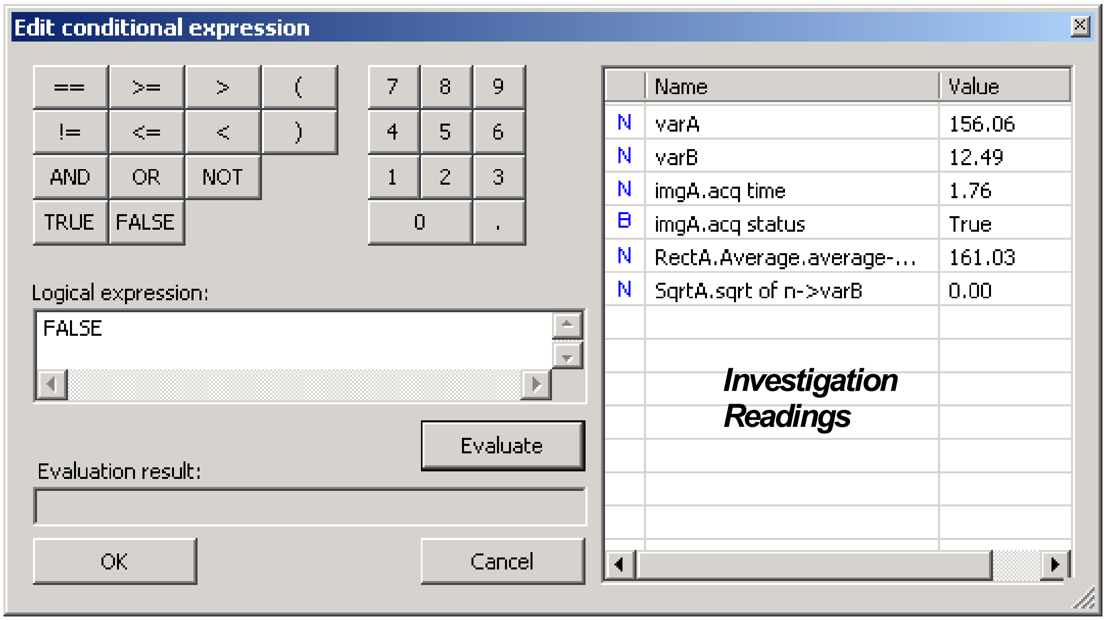
FALSE

Figure 30. ROI Tolerance

- *Enable/Pass/Tail Tolerance* - click in the check box to enable. Clear the check box to disable.
- *Condition* - displays the current Conditional Expression or test condition. Click the "Edit" button to change this expression (see page 46).
- *Edit* - opens the Edit Conditional Expression window.

Conditional Expression

The Conditional Expression is where you test limits on Sherlock readings. (In Sherlock 6 you entered tolerance numbers using the Edit Readings dialog.) You can test more than one limit, or more than one reading in one conditional expression. See the Tutorial, starting on page 31.



Edit conditional expression

Logical expression:

FALSE

Evaluation result:

	Name	Value
N	varA	156.06
N	varB	12.49
N	imgA.acq time	1.76
B	imgA.acq status	True
N	RectA.Average.average-...	161.03
N	SqrtA.sqrt of n->varB	0.00

Figure 31. Edit Conditional Expression

Operators and Numbers - Use the buttons to insert operators and numbers in the "logical expression" field. You can also type in the field.

Logical Expression - create an expression by typing, using the operators and numbers buttons, and draging readings into this field to create a test condition to pass or fail.

Readings - Drag readings into the "logical expression" field. All readings created in the investigation are listed in the right pane. As you create more readings through instructions or algorithms, they will appear in this list.

Evaluate - click to see the current result (True or False) displayed in the "Evaluation result" field.

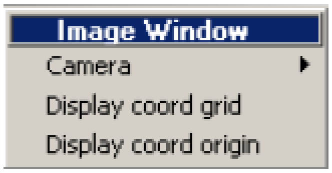
OK - accept changes to the logical expression, and close dialog.

Cancel - discard changes to the logical expression, and close dialog.

NOTE *You can also create a Conditional Expression by double-clicking on an* "If", "If-Else" *or* "While" *statement in the Program window.*

Image Window Options

Right-click in the image window (outside any ROI) to open a pop-up. This allows you to quickly change the camera if your hardware supports more than one camera. This pop-up also allows you to display or hide the coordinate grid, and the origin axis for measurements.



See Figure 6, page 23 for an illustration of the Main Window, Image Window, and Program Window.

To open the Image Window Options dialog:

- click on the Options button in the Image window toolbar,
- double-left-click in the image window (outside any ROIs),
- pull down the Image Window menu and select Options.

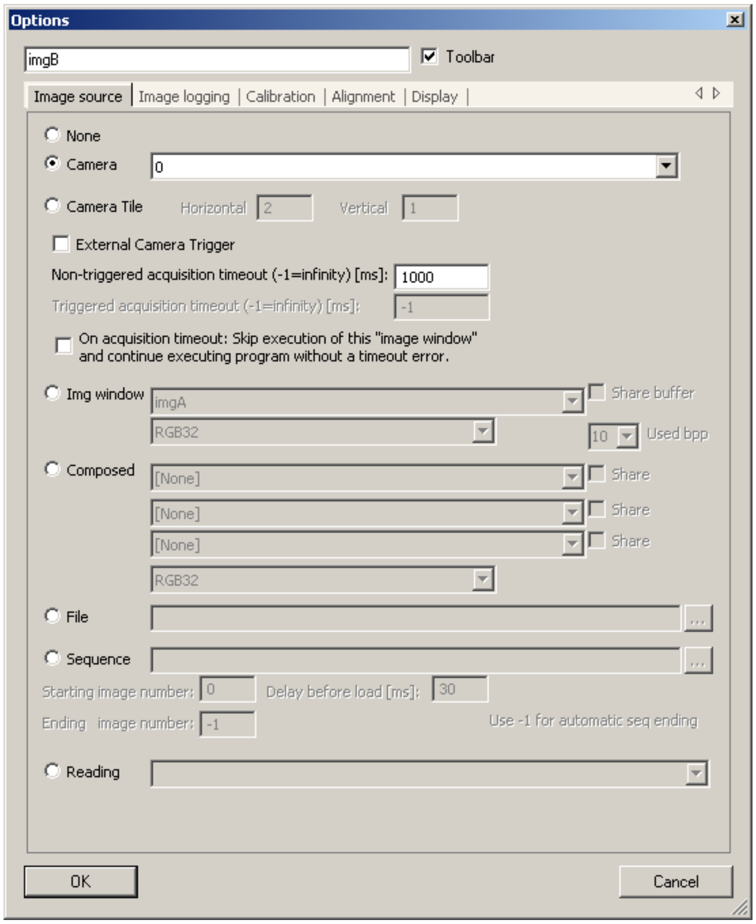
image window name - You can rename the image window by typing a new name in the text field at the top of this dialog. You can also rename the image window in the Program window by clicking on the name to highlight it, then press F2 (rename) and type a new name.

Toolbar - you can hide or show the image window toolbar, independently for each image window.

Tabs - This dialog has five tabs at the top for: Image source, Image logging, Calibration, Alignment, and Display

Image Window - Source Tab

Select where the image window gets its image from



Options

imgB ☑ Toolbar

Image source | Image logging | Calibration | Alignment | Display |

○ None

● Camera 0

○ Camera Tile Horizontal 2 Vertical 1

□ External Camera Trigger

Non-triggered acquisition timeout (-1=infinity) [ms]: 1000

Triggered acquisition timeout (-1=infinity) [ms]: -1

□ On acquisition timeout: Skip execution of this "image window" and continue executing program without a timeout error.

○ Img window imgA □ Share buffer

RGB32 10 Used bpp

○ Composed [None] □ Share

[None] □ Share

[None] □ Share

RGB32

○ File

○ Sequence

Starting image number: 0 Delay before load [ms]: 30

Ending image number: -1 Use -1 for automatic seq ending

○ Reading

Figure 32. Image Source

None - there is no image in this window. This option is used when an image is later created or loaded by the program.

Camera - a live camera is the image source. Use the pull down or list box to select a camera port supported by the installed hardware. You must have a valid software license for Sherlock to acquire from cameras. You can specify a timeout period to end waiting for a camera image. Sherlock will normally stop (halt) the investigation and report a timeout error.

Camera Tile - multiple camera images can be "tiled" into one image window.
Horizontal: number of camera images in the horizontal direction.
Vertical: number of camera images in the vertical direction.

- *External trigger:* Click the check box if you are using an external trigger signal. The image window, and Sherlock investigation will wait for a trigger pulse at the frame grabber or camera, to start acquisition. The camera is internally triggered (software trigger) if you do not check this box.
- *Non-Triggered Acquisition Timeout:* You can specify a timeout for image acquisition (internal timer or software triggered). This will cause Sherlock to register a timeout if the camera does not acquire an image. The timeout is reported as an error, unless the checkbox "On Acquisition Timeout . . . " is checked. Normally Sherlock halts on a timeout error, but this can be disabled in Application Options (Halt on camera timeout, page 61).

- *Triggered Acquisition Timeout:* You can specify a timeout for an external trigger image acquisition. This will cause Sherlock to register a timeout if the camera does not acquire an image. The timeout is reported as an error, unless the checkbox "On Acquisition Timeout . . . " is checked. Normally Sherlock halts on a timeout error, but this can be disabled in Application Options (Halt on camera timeout, page 61).
- *On Acquisition Timeout: skip execution of this Image Window and . . . without a timeout error:* If this box is not checked, the acquisition timeout (camera timeout) is logged as an error. Sherlock normally halts on a camera timeout (Acquisition Options, page 65). If this box is checked, Sherlock execution will skip all instructions and processing under this Image Window, and proceed to the next program step outside the scope of this image window instruction. With the timeout set to a low value, this creates a polling condition. This is useful when you have two externally triggered asynchronous cameras.

NOTE *If the acquisition timeout is set to -1 (infinity) and acquisition does not complete, you will have to terminate Sherlock with the Windows Task Manager.*

NOTE *A Subroutine that "Executes on camera timeout" (page 20) is executed as soon as the timeout error occurs, before the investigation may be terminated by "Halt on camera timeout". If the "On Acquisition Timeout . . ." box is checked, the image window does not generate a timeout error, and a subroutine that "Executes on camera timeout" is not executed.*

NOTE *When a triggered acquisition timeout occurs, the ROI(s) in the image window are executed if both "Halt on camera timeout" setting and "On Acquisition Timeout:" are disabled. You can use the "Acq Status" output to determine if an image was acquired or not. The ROI(s) are not executed when acquisition timeout, and either "halt on camera timeout" or "On Acquisition Timeout:" is enabled.*

Image window - this image window can get its image from another image window. This option is commonly used when you do not want all the ROIs and graphics or alignment targets displayed all in one window. Use the list box to select another image window. Select the image type (RGB, YUV, Mono8, Mono16) for Mono16, you can select the bit-depth used in this window.

- *Share buffer:* If you select "share buffer" the two image windows actually share the same physical memory, and the processing of one window shows up in the next window. If you do not select this option, Sherlock makes a copy of the camera image, and only the processing in the current window is displayed.

Composed - you can create or compose a 24-bit color image from two or three 8-bit images.

File - you can use a single image file. This is often useful when developing an investigation, experimenting with image tools, or testing images supplied to you as files.

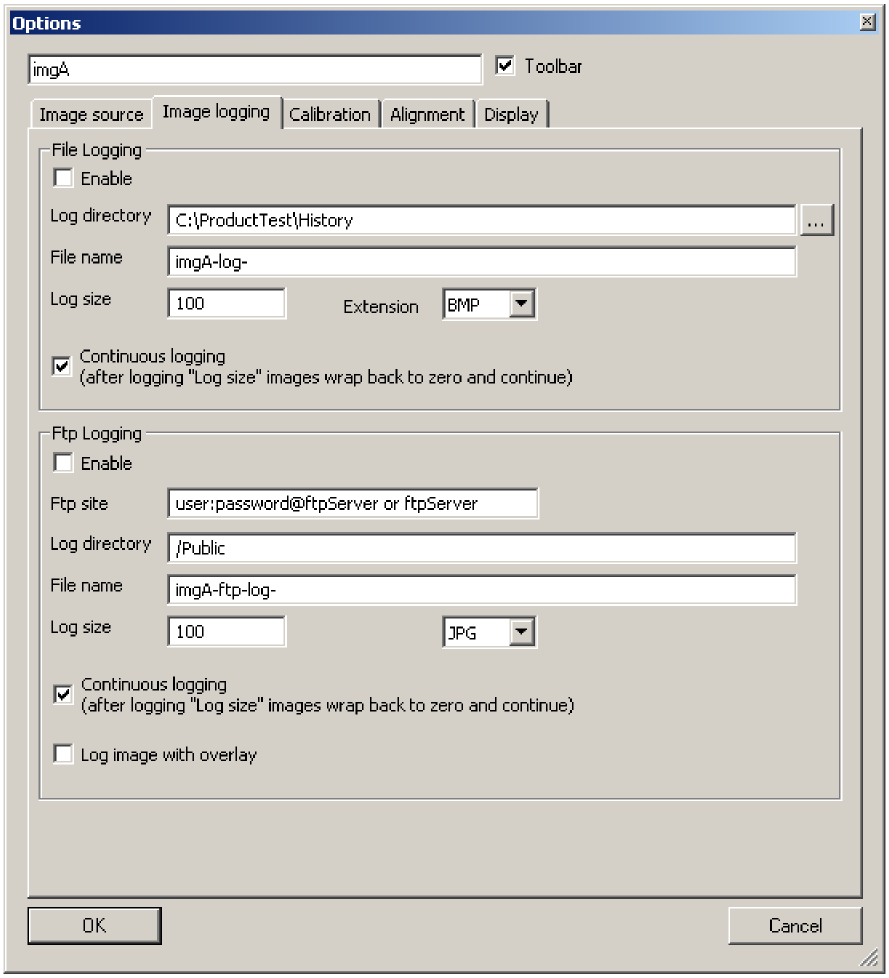
Sequence - you can use a sequence of saved images. This may be the result of image logging. At the end of the sequence, Sherlock will "wrap" to the first image and continue.

- *Starting image index:* You can start on any image in the sequence.
- *Ending image index:* end or loop on any image (-1 means use all the images, no limit).
- *Delay before load:* set the delay between images.

Reading - use a reading as input. A few algorithms output an image; for example, the Color Map algorithm outputs a gray-scale monochrome image.

Image Window - Image Logging Tab

Image Logging (Figure 33) saves a series of camera images. There are two options that operate independently. If you are using both options, the log directory path in the File Logging should not be the same as the temporary path created by FTP Logging.



imgA ☑ Toolbar

File Logging

☐ Enable

Log directory: C:\ProductTest\History

File name: imgA-log-

Log size: 100 Extension: BMP

☑ Continuous logging (after logging "Log size" images wrap back to zero and continue)

Ftp Logging

☐ Enable

Ftp site: user:password@ftpServer or ftpServer

Log directory: /Public

File name: imgA-ftp-log-

Log size: 100 JPG

☑ Continuous logging (after logging "Log size" images wrap back to zero and continue)

☐ Log image with overlay

Figure 33. Image Logging

File Logging

The "Image Log" is a sequence of saved camera images. You can save images and later use the "Image Log" as an image source to a Image Window.

Enable - click the check box to enable logging, or saving images.

Disable automatic numbering - (7.2.0.8 and later) normally a 4-digit number is added to your path and file name. When you disable automatic numbering, you can supply an image file number by using a variable. Disabling automatic numbering also disables the **Log size** and **Continuous logging** features. You must manage the log size by limiting or constraining your variable's values. If you do not supply an appropriate variable value, a single image file will be continuously overwritten.

Log directory - type a directory path or use the browse button to the right of the text field to select a directory.

File name - enter a file name prefix. The software appends a four-digit number to the prefix. The number will automatically expand when the count exceeds four digits.

Log size - enter the maximum number of images to save (for this image window). If **Continuous logging** is enabled, the index numbers used to save images are 0 to ("log size" minus 1) and the index counter wraps or resets back to zero and begins overwriting previous images. If **Continuous logging** is not enabled, the index numbers used are 0 to "log size" and no more images are saved after the "log size" index is reached. The maximum entry value is 2,147,483,648 (-1 is not supported).

Extension - select a file format (image file type): BMP, JPG, PNG, TIF.

Continuous logging - if this box is checked, Sherlock will start overwriting the first image captured when the maximum number “log size” is reached. If you do not select this option, Sherlock will stop saving images when the “log size” is reached.

Variable substitution (expansion) syntax is supported for directory and file name prefix. Variable names inside square brackets are evaluated and passed to the logging thread.

There is no parsing or error checking at run time. If the directory does not exist, or the prefix is not valid, the image will not be logged (not saved) and there will not be an error message.

We recommend you do not use the same drive partition as your OS (usually the D: drive) for logging images. A flaw in the Windows NTFS file system can cause corruption of the boot sector under continuously writing files such as images to the OS partition.

FTP Logging

The “FTP Log” is a sequence of saved camera images. These images are saved to an FTP server. You must create a shared directory on the FTP server for software Sherlock to save images in. Each image window can create a separate FTP thread for logging images. The “File name” must be different for each image window.

Enable - click the check box to enable logging, or saving images.

Disable automatic numbering - (7.2.0.8 and later) normally a 4-digit number is added to your path and file name. When you disable automatic numbering, you can supply an image file number by using a variable. Disabling automatic numbering also disables the **Log size** and **Continuous logging** features. You must manage the log size by limiting or constraining your variable’s values. If you do not supply an appropriate variable value, a single image file will be continuously overwritten.

FTP Site - the FTP Address of the server if logging in is not required, or Username:password @FTP address.
For example, MyName:SecretPassword@ftp.teledynedalsa.com
Variable expansion is not supported in this field.

Log directory - type a full path to your shared directory on the FTP server. The path should start with a forwarded slash, and may be case sensitive. Do not put a forwarded slash at the end of the path.
For example, /Public/Engineering/ImageLog

File name - enter a file name prefix. The software appends a four-digit number to the prefix. The number will automatically expand when the count exceeds four digits.

Log size - enter the maximum number of images to save (for this image window). If **Continuous logging** is enabled, the index numbers used to save images are 0 to (“log size” minus 1) and the index counter wraps or resets back to zero and begins overwriting previous images. If **Continuous logging** is not enabled, the index numbers used are 0 to “log size” and no more images are saved after the “log size” index is reached. The maximum entry value is 2,147,483,648 (-1 is not supported).

Extension - select a file format (image file type): BMP, JPG, PNG, TIF.

Continuous logging - if this box is checked, Sherlock will start overwriting the first image captured when the maximum number “log size” is reached. If you do not select this option, Sherlock will stop saving images when the “log size” is reached.

Log image with overlay - include all overlay and annotations in the image. This option increases the file size and the data transfer time.

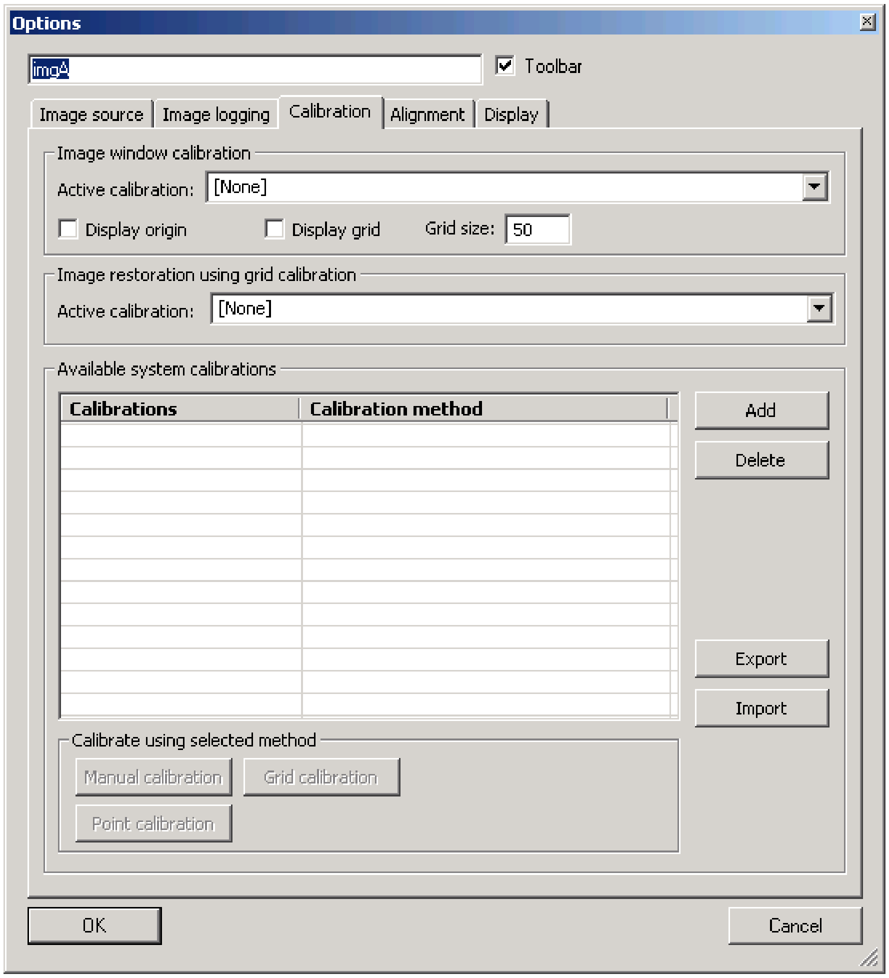
Variable substitution (expansion) syntax is supported for directory and file name prefix. Variable names inside square brackets are evaluated and passed to the logging thread.

There is no parsing or error checking at run time. If the directory does not exist, or the prefix is not valid, the image will not be logged (not saved) and there will not be an error message.

Images are saved in a temporary buffer or queue on your PC or Vision Appliance before transfer. If you get a message that images are being skipped, “Queue maximum reached. Skipping image ##”. Increase the size of the queue in Options➔Application settings; Advanced settings: Number of images queued by the FTP logging.

Image Window - Calibration Tab

Calibration changes distances measured in pixels into real units of measure. Some methods also correct for distortions. Some methods allow you to use the same image for calibration, and measurements. For other methods, it is better to use one "target" image for calibration, and a different image for measurements. The target image should be placed in front of the camera, at the same distance as the images that will be measured.



Options

imgA ☑ Toolbar

Image window calibration

Active calibration: [None]

□ Display origin □ Display grid Grid size: 50

Image restoration using grid calibration

Active calibration: [None]

Available system calibrations

Calibrations	Calibration method

Figure 34. Calibration

Image window calibration

Active calibration - use the drop list (or "combo box") to select a defined calibration to apply to this image window. All defined calibrations are listed in the field below. If you leave this box "None" the image window is not calibrated and all measurements are in pixels. Some methods for creating a calibration are on the following pages. See also Camera Calibration beginning on page 88.

Display origin - enable or disable displaying the X,Y origin in this image window.

Display grid - enable or disable displaying a grid in this image window. This is the "measurement grid" in pixels or calibrated units.

Grid size: enter a number of pixels (or calibrated units) between grid lines.

Image restoration using grid calibration

Active calibration - select the (already defined) grid calibration to use in redrawing this image window, displayed with corrections or "restoration." You do not need to apply restoration to make calibrated measurements. You do need to apply restoration before you can use a search algorithm.

Available system calibrations

Calibrations - lists the calibration systems already defined, and the method used. You can create a calibration system using instructions in the program window, or use the buttons on this menu.

Add - Click to add a new calibration system. You can add a calibration here, and select it in the program window. Or you can add a calibration and use the buttons below to calibrate this image window. You must **Add** a calibration first, before you can run calibrate.

Delete - Select a calibration system in the list, and click to delete the calibration system from the investigation.

Export - Save the calculated calibration data for this investigation, to a file (*.cal). This data is only valid for the same camera, same lens, same camera location, and distance to object.

Import - Load the previously saved or Exported calibration data. This is only valid when using the same camera, same lens, same camera location, and same distance to the object.

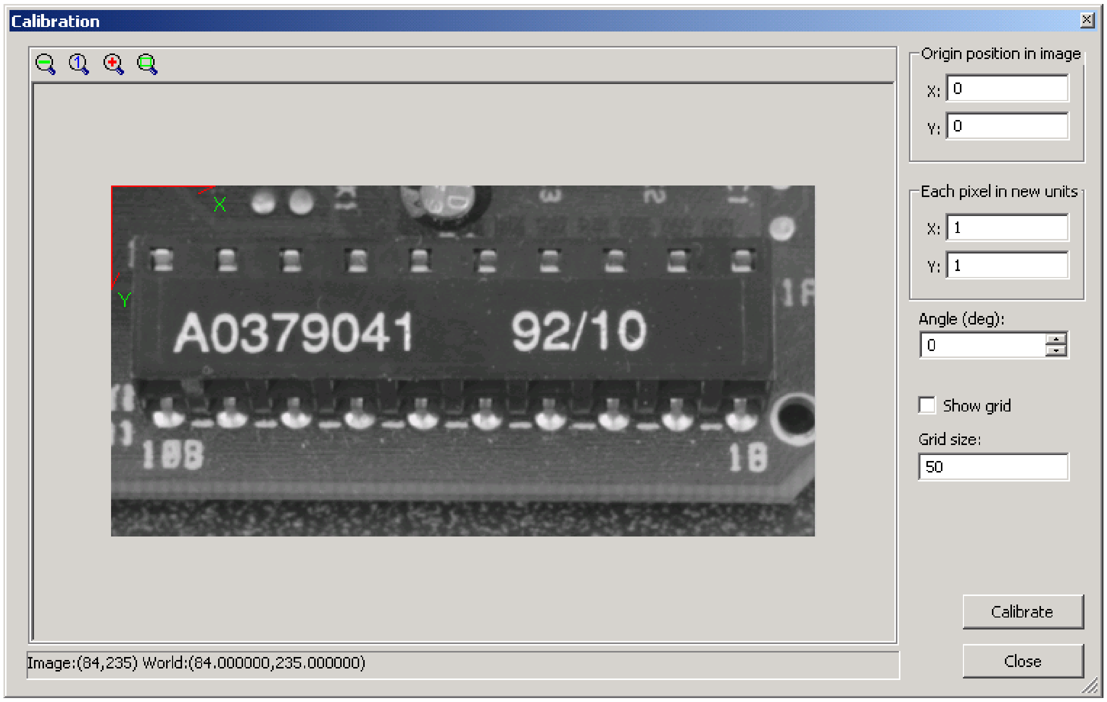
Manual calibration - Select a calibration system in the list, and click this button to define the calibration based on multipliers or scaling ratios. Clicking this button opens a Calibration window. The calibration image (target) must be in the Image Window before clicking this button. See also Camera Calibration on page 88.

Point calibration - Select a calibration system in the list, and click this button to define the calibration based on points you create in the image. Clicking this button opens a Calibration window. The calibration image (target) must be in the Image Window before clicking this button. See also, Camera Calibration (page 88).

Grid calibration - Select a calibration system in the list, and click this button to define the calibration based on a grid or checkerboard. Clicking this button opens a Calibration window. The calibration image (grid) must be in the Image Window before you click this button. See also, Camera Calibration (page 88).

Manual Calibration

Manually calibrate distance measurements, by providing scale (or multiplication) factors. This method is very useful if you know the exact size of an object filling the camera image, or the exact size of the field of view.



Origin position in image
X: 0
Y: 0

Each pixel in new units
X: 1
Y: 1

Angle (deg): 0

☐ Show grid

Grid size: 50

Image:(84,235) World:(84.000000,235.000000)

Figure 35. Direct Calibration

Origin position in image - enter x and y coordinates for the measurement origin. This defines positive and negative directions. The default is 0,0 the top left of the image. Click "Calibrate" or change the angle to see the origin move to the new position.

Each pixel in new units - enter your multiplication factors for x and y. The default is 1,1 which gives measurements in pixels. For example; if you know the camera image is 5 cm wide, and the Image window title bar says the image is 648x489:

5 cm / 648 pixels = 0.00771 cm per pixel. Use 0.00771 for the X value. If the number gets rounded up, check the setting of "Displayed floating point precision" in Application options.

Angle (deg) - use the arrows or enter a number, to rotate the X,Y axis, to define the 0 rotation for your measurements.

Show Grid - click the checkbox to display the measurement grid.

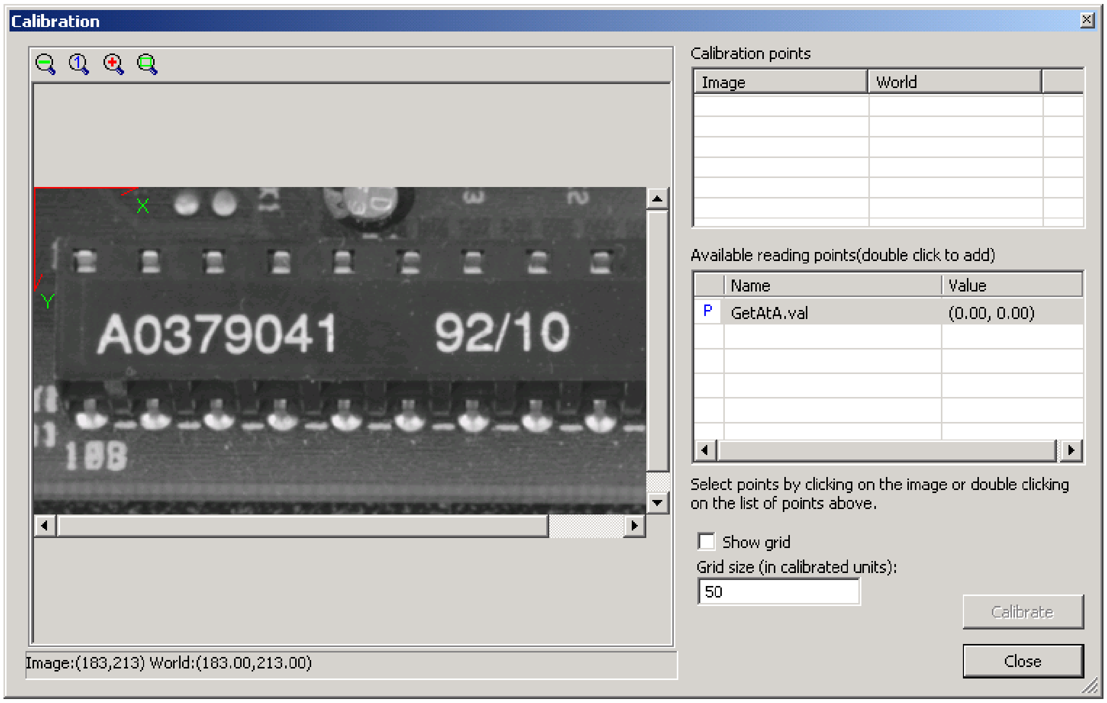
Grid Size: enter a number for grid size (in calibrated units), then click in the checkbox. If the old grid is displayed, click Show Grid once to hide the old grid, and click a second time to display the new grid.

Calibrate - click the button to execute your new calibration scale factors and rotation.

Close - closes the Calibration window.

Point Calibration

Manually calibrate distance measurements, by supplying points for reference. This method is very useful if you know the exact size of an object in the camera image. You need four points for calibration. Usually the four points are near the corners of the image or the object. One is taken as point 0,0. The others points are the x and y distances (in inches, millimeters, thousandths) from the first point.



Name	Value
P GetAtA.val	(0.00, 0.00)

Select points by clicking on the image or double clicking on the list of points above.

Grid size (in calibrated units): 50

Image:(183,213) World:(183.00,213.00)

Figure 36. Point Calibration

This window provides two ways to select points for calibration:

- "Available Points" lists points created by your investigation; usually the results (readings) of processing. Double-click on an available point listed, and enter a "world position" in the pop-up. Click OK and the point is listed in the "Calibration points".
- Click on the image to create a point, and enter a "world position" in the pop-up. Click OK and the point is listed in the "Calibration points".

Calibration points - lists points you have assigned to the calibration.

- *Image:* point coordinates in pixels.
- *World:* point coordinates in new units.

Available reading points - lists all point readings defined in the investigation.

Show grid - enable grid display, in the image in this dialog only.

- *Grid size:* separation or spacing of grid lines, in calibrated units.

Calibrate - click the button to execute your new calibration.

Close - closes the Calibration window.

Grid Calibration

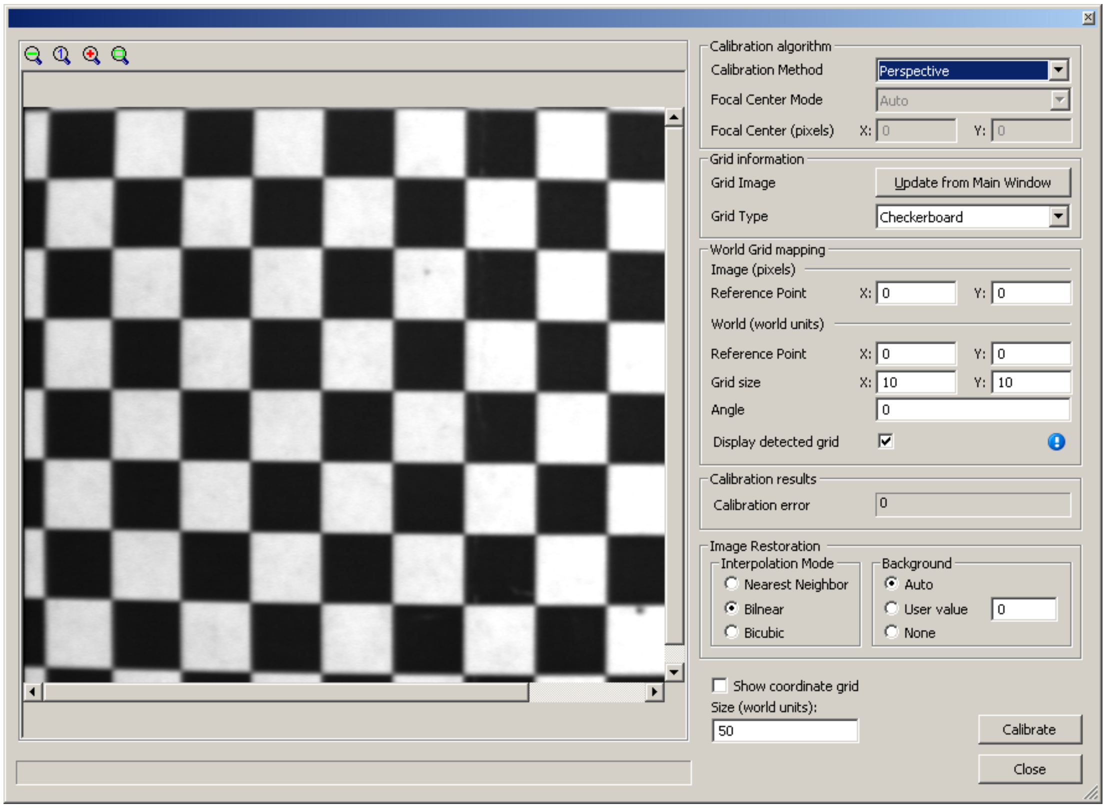
Manually (live or interactive) calibrate distance measurements using: a checkerboard, grid of lines, or grid of spots. This method is useful if you know the separation of elements in the grid. This method is also very useful for correcting perspective or radial distortion

NOTE *For best results, the image grid or checkerboard should fill the image window.*

Calibration Algorithm

Calibration Method - select the type of distortion present in the image. Use Perspective if you do not need correction, because the least amount of calculation is involved.

- *Piecewise Perspective:* corrects for localized and perspective distortions. Four point mapping is calculated for each square or cell of the grid.
- *Piecewise Bilinear:* corrects for localized distortions, but not for perspective distortion. Assume the image plane is parallel to the camera sensor plane (see page 95). Bilinear mapping is calculated for each square or cell in of the grid.
- *Radial:* corrects for radial distortion only. Does not correct for perspective distortion. Radial distortion is a common deformation of some lenses; “fish eye” or “barrel” distortion are common names for some of these.
- *Perspective:* corrects for perspective distortion only. One four point mapping is calculated for the whole image. This is the fastest calibration method; using fewer calculations.



Calibration algorithm

Calibration Method: Perspective

Focal Center Mode: Auto

Focal Center (pixels) X: 0 Y: 0

Grid information

Grid Image

Grid Type: Checkerboard

World Grid mapping

Image (pixels)

Reference Point X: 0 Y: 0

World (world units)

Reference Point X: 0 Y: 0

Grid size X: 10 Y: 10

Angle: 0

Display detected grid ☑

Calibration results

Calibration error: 0

Image Restoration

Interpolation Mode: ○ Nearest Neighbor ● Bilnear ○ Bicubic

Background: ● Auto ○ User value 0 ○ None

☐ Show coordinate grid

Size (world units): 50

Figure 37. Grid Calibration

NOTE *All 4 methods use the same amount of time at inspection runtime. Some methods take longer to calculate the remapping at Calibration time.*

Focal Center Mode (active if Calibration Method is Radial) - define how the center of the radial distortion is found:

- *Auto:* the algorithm automatically finds or calculates the center of radial distortion.
- *Fixed:* the center point of radial distortion is entered manually, as X and Y pixel coordinates.

Focal Center (pixels) (active if Calibration Method is Radial, and Focal Center is Fixed) - the center of radial distortion is entered as X and Y pixel coordinates.

Grid Information

Update from Main Window - Click to update this display from the Image Window. If you have run calibration before, the last calibration image may be displayed in the image area.

Grid Type - define the target or calibration image. Generally, a checkerboard usually gives the best results, a spot or dot pattern gives the least accurate results. Sometimes the application or physical limitations may require using an inferior pattern.

- *Checkerboard:* reference grid is formed by alternating black and white squares.
- *Lines:* grid is formed by intersecting perpendicular lines.
- *Spots:* grid is formed by spots (or perforations) spaced uniformly at horizontal and vertical intervals.
- *Black Lines:* grid is formed by intersecting perpendicular black lines on a white background.
- *Black Spots:* grid is formed by black spots on a white background, spaced uniformly at horizontal and vertical intervals.
- *White Lines:* grid is formed by intersecting perpendicular white lines on a black background.

- *White Spots:* grid is formed by white spots on a black background, spaced uniformly at horizontal and vertical intervals.

World Grid mapping

Image (pixels): Reference Point - the pixel location in the image, used as the origin for measurements. Run the calibration once, without entering a reference point. Then click in the calibration image to define the point of origin. The number fields are filled when you click on the image area, with the un-calibrated pixel coordinates. Run calibration a second time, to use the same grid and measurements, but set the new origin point.

World (world units) Reference Point - the location in world or calibrated units, or the origin; defines positive and negative directions.

Grid size - known distance or spacing in the grid. Enter horizontal (X) and vertical (Y) spacing.

Angle - if the grid is at a known angle from the horizontal, enter a value (default is in degrees).

Display detected grid - if checked, green lines are drawn in the Calibration window to indicate the resulting distortion grid area. If the grid does not fill the image, these lines may become confused outside the grid, using some correction methods.

Calibration results

Calibration error - sum of the distance errors in calculating distortion corrections. Zero is the ideal number. You can use this value to compare repeated calibrations. But, increasing the number of cells or squares from one calibration to the next increases the number of distanced used, and makes comparison not valid. *Calibration error does not apply to the Piecewise methods.*

Image Restoration

Interpolation Mode - how to create new pixels in the restoration image, that do not correspond to pixels in the original image; used to fill spaces.

- *Nearest neighbor:* use the value of the nearest pixel (no interpolation). This is the fastest but least accurate.
- *Bilinear:* calculate bilinear interpolation using the four surrounding pixels from the original image. This is more accurate and slower than Nearest neighbor, but not as accurate or as slow as "Bicubic" interpolation.
- *Bicubic:* calculate bi-cubic interpolation using the sixteen surrounding pixels from the original image. This is the most accurate but also the slowest method of interpolation.

Background - how to generate pixels for areas outside the restoration image. There is very little difference in the speed of these three methods.

- Auto: average the pixel values at the four corners of the original image. This method is very slightly slower.
- User value: use the pixel value entered in the number field.
- None: do not create a new background. Use the original image, and overlay the result on top of the original.

Show coordinate grid - display the measurement grid.

Size (world units): - spacing of coordinate grid, in calibrated units.

NOTE *Calibration and Restoration are separate operations, and cannot be done at the same time. Perform a calibration in one Image Window. Select the completed calibration as Active, for performing restoration in a different Image Window. The two windows can use the same image source.*

Grid Calibration Targets

Three very important limits on the grid calibration and targets are discussed here: minimum number of points, buffer or edge zone, origin and calibration.

- The minimum number of grid points is **four**, which is acceptable for pure perspective distortion, forming a square. However, the greater the number of grid points, the better the calibration; therefore it is recommended that the most grid points possible be used.

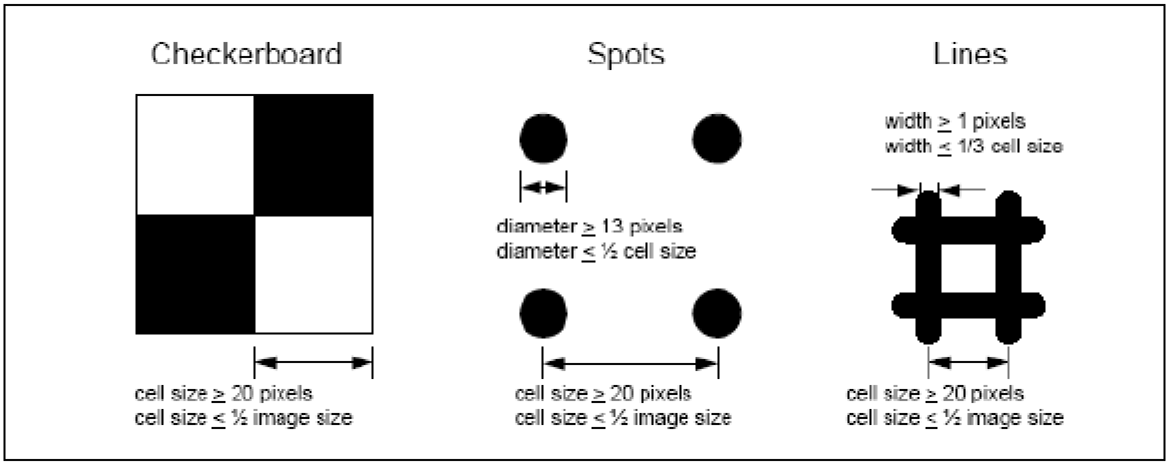
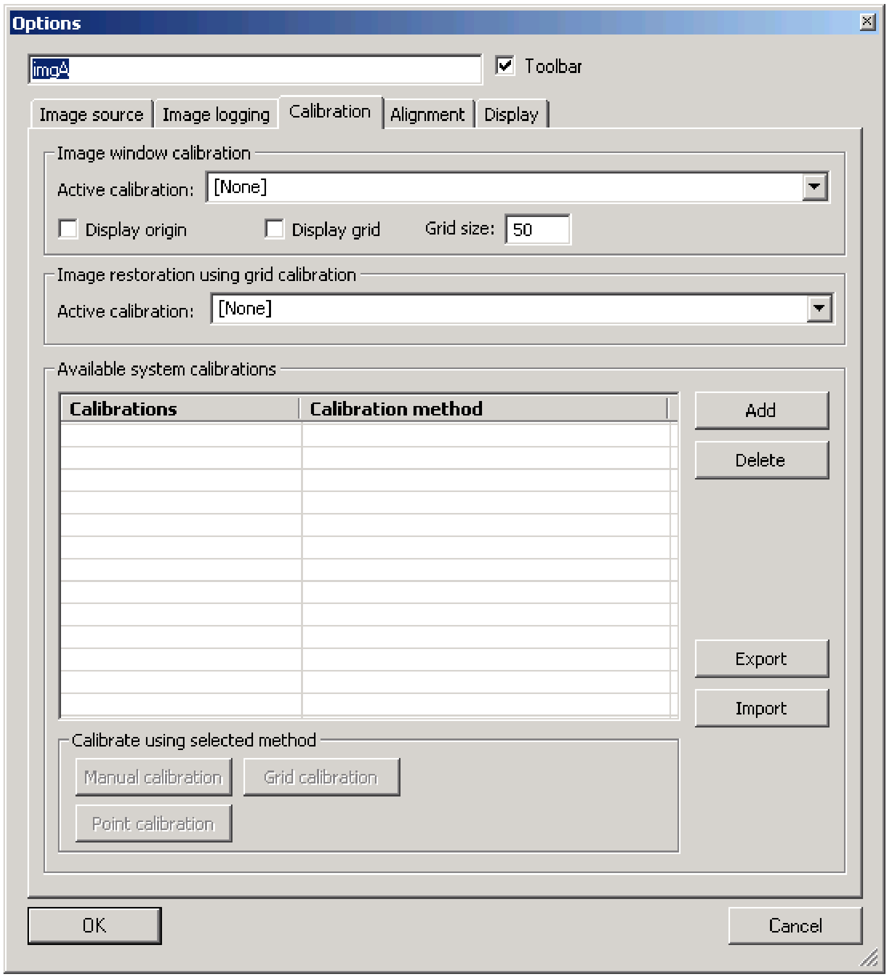


Figure 38. Minimum Grid Targets

- For the checkerboard or line calibration targets, the algorithms require a **10-pixel** buffer between any point on an edge and the edge of the image, for the square to be used in the calibration. For example, if the top edge of a square in a checkerboard is only 8 pixels from the top edge of the image, the square will not be used to create the correction map, even though the entire square is in the image.
- Grid calibration corrects the source image starting at the point designated as the origin in the detected grid display in the calibration dialog. By default, this is the upper-leftmost spot center or square corner in the grid; it is marked with two red arrows. This origin becomes the (0,0) pixel of the corrected image; pixels to the left and above the detected grid origin are ”mapped out” of the corrected image.

Image Window - Alignment Tab

Sherlock allows you to create multiple alignment systems, for each camera or each image window. You create and manage the alignment systems here. Refer to Alignment beginning on page 96. **You select and assign an alignment system in the ROI properties**



imgA ☑ Toolbar

Image window calibration

Active calibration: [None]

□ Display origin □ Display grid Grid size: 50

Image restoration using grid calibration

Active calibration: [None]

Available system calibrations

Calibrations	Calibration method

Calibrate using selected method

Figure 39. Alignment

Alignments - lists the current defined alignment systems. Use the “Create alignment” button to create a new alignment system. You can rename the alignment, by slowly clicking twice on the name, and typing a new name in the active field.

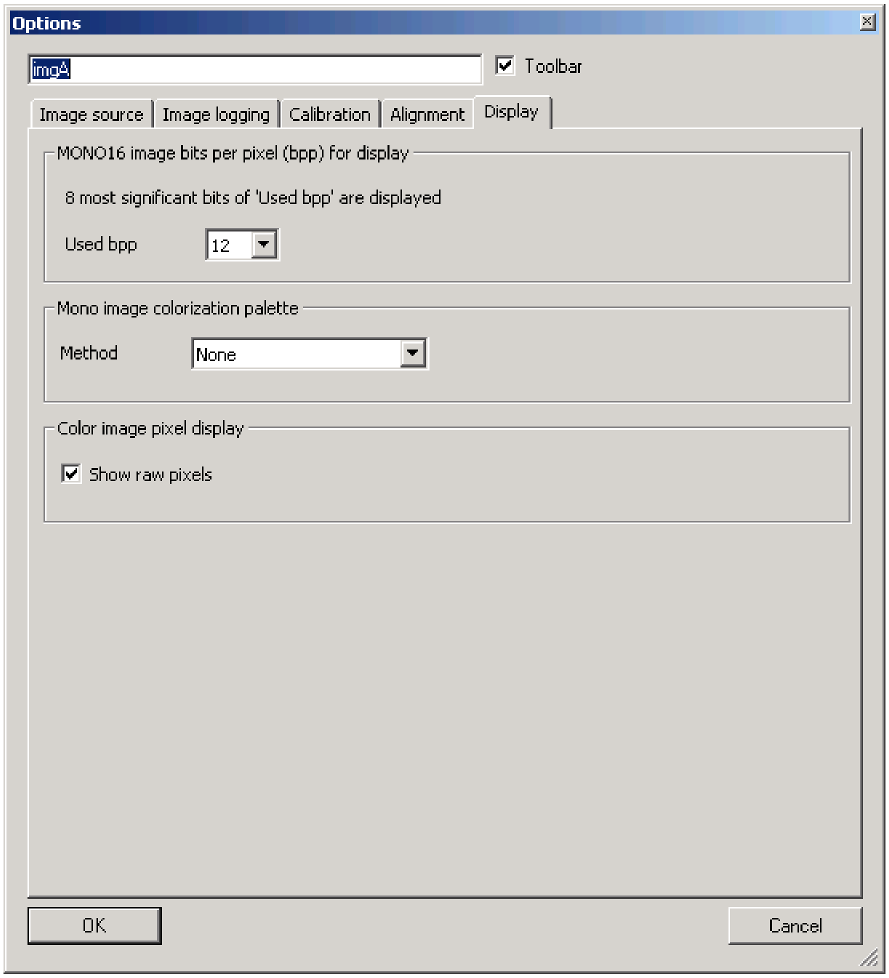
Point readings - lists all the point readings available in the investigation. Points created by your investigation; usually the results (readings) of processing. You drag point readings and drop them on a alignment system name. Points can be used in more than one alignment system.

Create alignment - click to add a new name to the list of alignments. Drag and drop point readings onto the alignment name.

Reset home position - Select an alignment in the list, and click this button to store this alignment and make this alignment the new beginning reference point when creating new alignment systems.

Image Window - Display Tab

Bpp for 16-bit images - Monochrome images that are greater than 8 bits per pixel, are assigned type MONO16. The display option allows you to set the actual pixel size, so the image is correctly displayed in the image window. Use the drop-list to select the pixel "depth" or number of bits.



Options

imgA

☑ Toolbar

MONO16 image bits per pixel (bpp) for display

8 most significant bits of 'Used bpp' are displayed

Used bpp: 12

Mono image colorization palette

Method: None

Color image pixel display

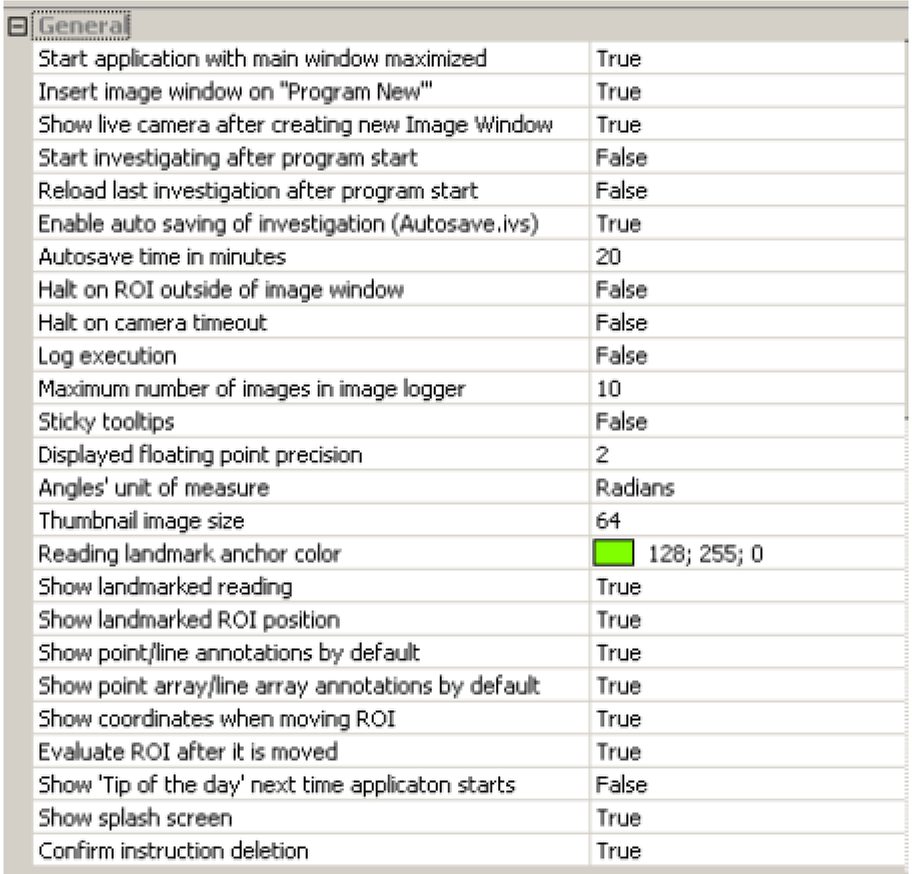
☑ Show raw pixels

Figure 40. Display

Application Options

Open the Applications Options from the Options menu: Options ➔ Applications.

Configure how Sherlock opens and runs, some of the display defaults and colors. These settings are applied to all inspection programs on this PC. **Note:** The Help file may show more up-to-date information and options.



⊟ General

Start application with main window maximized	True
Insert image window on "Program New"'	True
Show live camera after creating new Image Window	True
Start investigating after program start	False
Reload last investigation after program start	False
Enable auto saving of investigation (Autosave.ivs)	True
Autosave time in minutes	20
Halt on ROI outside of image window	False
Halt on camera timeout	False
Log execution	False
Maximum number of images in image logger	10
Sticky tooltips	False
Displayed floating point precision	2
Angles' unit of measure	Radians
Thumbnail image size	64
Reading landmark anchor color	🟩 128; 255; 0
Show landmarked reading	True
Show landmarked ROI position	True
Show point/line annotations by default	True
Show point array/line array annotations by default	True
Show coordinates when moving ROI	True
Evaluate ROI after it is moved	True
Show 'Tip of the day' next time applicaton starts	False
Show splash screen	True
Confirm instruction deletion	True

Figure 41. Application Options - General

General:

Start application with main window maximized - start Sherlock with the Main window open (maximized) or minimized.

Insert image window on "Program new" - automatically inserts a image window instruction when you select New from the File menu (create a new investigation).

Show Live Camera after creating new Image Window - default to a live image acquire in a new image window, or when opening Sherlock.

Acquire Image after program load - with live camera disabled, a single camera image is acquired when a investigation file is loaded, unless you disable this feature.

Start investigating after program start - starts running when you open Sherlock, after reloading the last investigation (if enabled). This setting is saved in the current investigation file (usually used with the previous option to automatically load the last investigation).

Reload last investigation after program start - reloads previous or last investigation when you open Sherlock. (usually used with the next option, to start testing after loading).

Enable auto saving of investigation (Autosave.ivs) - modified investigation files can be automatically saved on a set time period.

Autosave time in minutes - sets the time period for auto-saving modified investigation files.

Halt on ROI outside of image window - terminate the investigation if a ROI goes outside the image window. Alignment can make an ROI move outside the image window.

Halt on camera timeout - stops the investigation if the sensor does not complete the acquisition in the timeout period (set in the Image window options). This avoids hanging the application, computer, and peripherals.

Log execution - enable or disable logging execution of program instructions to a file “IpeStudioLog.txt” for debug purposes.

Maximum number of images in image logger - set the number of images kept in a buffer in memory, by the IO:Image Window:Log Image instruction. For debug use.

Sticky tooltips - with this option True, once a tooltip opens, you have to click to close it.

Displayed floating point precision - number of decimal points displayed.

Angles’ unit of measure - select Degrees or Radians as the default display for all angles.

Thumbnail image size - change the size of thumbnails when viewing “All” images in the Global Images window.

Reading landmark anchor color - assign the color of a landmark (alignment) RGB value.

Show landmarked reading - show or hide the landmark (alignment).

Show landmarked ROI position - show or hide original position of landmarked ROI.

Show point/line annotations by default - show or hide point and line annotations in the image window. Individual ROIs can override this setting.

Show point array/line array annotations by default - show or hide point and line arrays annotations in the image window. Individual ROIs can override this setting.

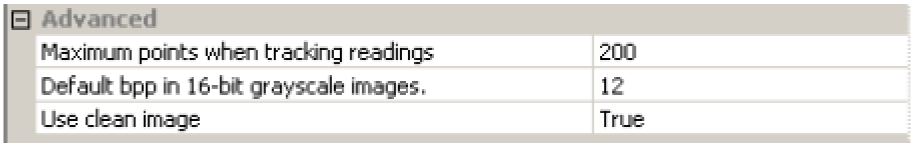
Show coordinates when moving ROI - coordinates will display at the top of the image window as you drag the ROI.

Evaluate ROI after it is moved - turn on/off reevaluating ROI when moved and released. If True, a new result is available when you release the ROI after dragging. If False, you must run the program to get new results.

Show “Tip of the day” next time application starts - you can turn off the Tip of the day.

Show splash screen - show or hide the Sherlock splash screen when you open Sherlock.

Confirm instruction deletion - you can disable or re-enable the message that requires you to confirm a deletion.



⊟ Advanced

Maximum points when tracking readings	200
Default bpp in 16-bit grayscale images.	12
Use clean image	True

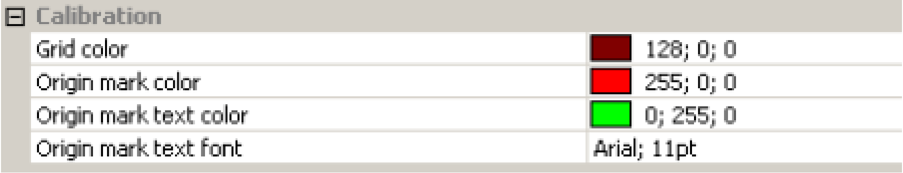
Figure 42. Application Options - Advanced

Advanced:

Maximum points when tracking readings - changes the X axis showing the readings, in graph displays.

Default bpp in 16-bit grayscale images - what the real image bit size is. This affects the display of images greater than 8 bits. This default can be overridden in each image window (display tab).

Use clean image - if True, each time a new image is acquired a clean image is cached for each image source and used when: ROIs are evaluated, image logging instruction executes, preprocessors are reordered/disabled/added/deleted, algorithms are added or deleted. These conditions are usually seen when editing the program. An investigation program must be reloaded, or Sherlock closed and restarted, to clear the cached images. If False, each image source has one less image allocated, and less memory is used. If an ROI is reevaluated, the result may be from the previous result, and not from the original acquired image.



⊟ Calibration	
Grid color	128; 0; 0
Origin mark color	255; 0; 0
Origin mark text color	0; 255; 0
Origin mark text font	Arial; 11pt

Figure 43. Application Options - Calibration

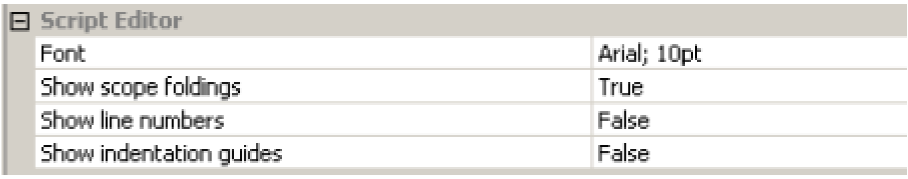
Calibration: (*all colors are given in RGB values*)

Grid color - select the color of the grid. Grid is enabled in the Image window options, calibration tab.

Origin mark color - select the color of the origin mark, the X,Y axis. Show/hide the origin is enabled in the Image window options, calibration tab.

Origin mark text color - select the color of the X and Y labels on the origin mark.

Origin mark text font - select the font and size of the X and Y labels on the origin mark.



⊟ Script Editor	
Font	Arial; 10pt
Show scope foldings	True
Show line numbers	False
Show indentation guides	False

Figure 44. Application Options - Script Editor

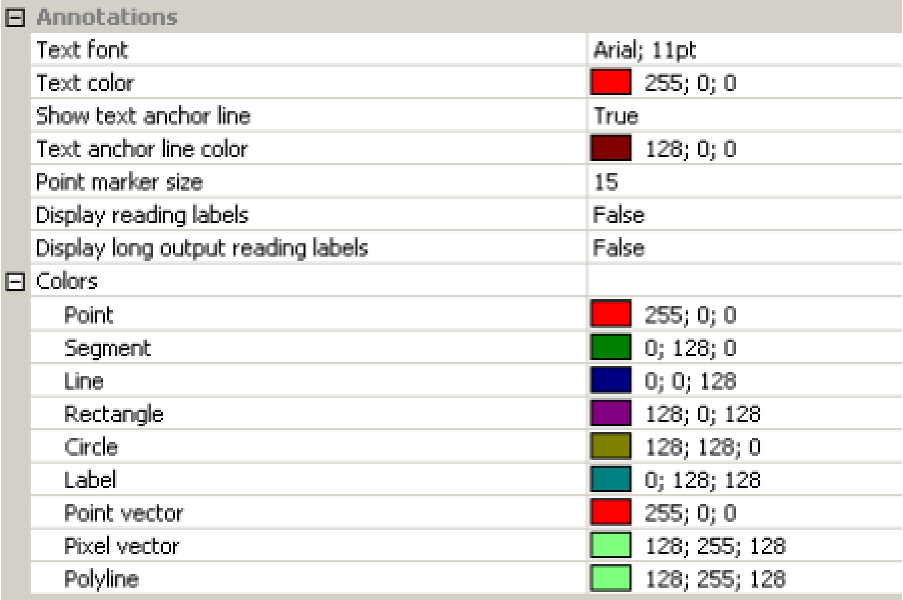
Script Editor:

Font - select the font and size used in the script editor.

Show scope foldings - enable expanding and collapsing (folding) code segments (scope) under certain statements. A javascript visual tool.

Show line numbers - show line numbers along the border of the script editor window.

Show indentation guides - use indentation to indicate hierarchy in the script editor.



⊟ Annotations	
Text font	Arial; 11pt
Text color	255; 0; 0
Show text anchor line	True
Text anchor line color	128; 0; 0
Point marker size	15
Display reading labels	False
Display long output reading labels	False
⊟ Colors	
Point	255; 0; 0
Segment	0; 128; 0
Line	0; 0; 128
Rectangle	128; 0; 128
Circle	128; 128; 0
Label	0; 128; 128
Point vector	255; 0; 0
Pixel vector	128; 255; 128
Polyline	128; 255; 128

Figure 45. Application Options - Annotations

Annotations: (*all colors are given in RGB values*)

Text font - select the font and size for annotations in the image window.

Text color - select the color of annotation text in the image window.

Show text anchor line - show or hide the line from the label to the point associated with it. Not all readings use anchor lines.

Text anchor line color - select the color for the anchor line.

Point marker size - select the size of the crosshairs that show a point reading (edges, centroids).

Display reading labels - display reading values in the image window.

Display long output reading labels - turn off abbreviation of reading labels in the image window.

Point - select the color of a point annotation.

Segment - select the color of a line segment annotation.

Line - select the color of a line annotation.

Rectangle - select the color of a rectangle annotation.

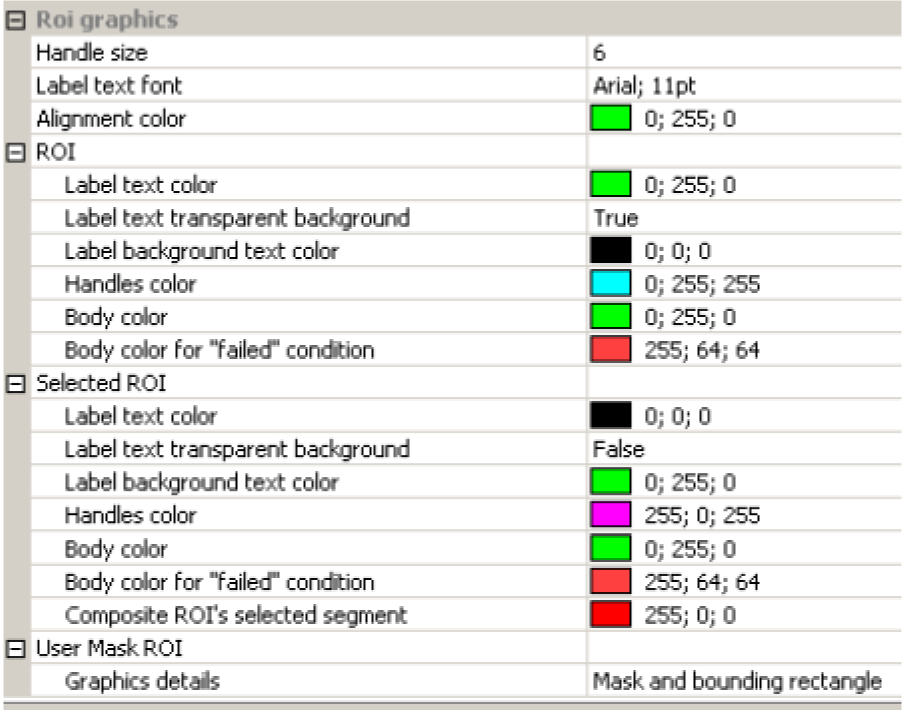
Circle - select the color of a circle annotation.

Label - select the color of a label annotation.

Point vector - select the color of a point vector annotation.

Pixel vector - select the color of a pixel vector annotation.

Polyline - select the color of a polyline annotation.



⊟ Roi graphics	
Handle size	6
Label text font	Arial; 11pt
Alignment color	🟩 0; 255; 0
⊟ ROI	
Label text color	🟩 0; 255; 0
Label text transparent background	True
Label background text color	⬛ 0; 0; 0
Handles color	🟦 0; 255; 255
Body color	🟩 0; 255; 0
Body color for "failed" condition	🟥 255; 64; 64
⊟ Selected ROI	
Label text color	⬛ 0; 0; 0
Label text transparent background	False
Label background text color	🟩 0; 255; 0
Handles color	🟪 255; 0; 255
Body color	🟩 0; 255; 0
Body color for "failed" condition	🟥 255; 64; 64
Composite ROI's selected segment	🟥 255; 0; 0
⊟ User Mask ROI	
Graphics details	Mask and bounding rectangle

Figure 46. Application Options - ROI Graphics

ROI Graphics: (*all colors are given in RGB values*)

Handle size - select the size of the drawing handles.

Label text font - select the font and size of ROI labels.

Alignment color - select the color of a Aligned ROI outline.

Label text color - select the color of a ROI label, when not selected.

Label text transparent background - show or hide a background on a text label, when not selected.

Label background text color - select the color of a label background, when not selected.

Handles color - select the color of the drawing handles, when not selected.

Body color - select the color of the ROIs outline, when not selected.

Body color for "failed" condition - select the color of the not-selected ROIs outline, when the ROI Tolerance conditional statement fails.

Label text color - select the color of a ROI label, when selected.

Label text transparent background - show or hide a background on a text label, when selected.

Label background text color - select the color of a label background, when selected.

Handles color - select the color of the drawing handles, when selected.

Body color - select the color of the ROIs outline, when selected.

Body color for "failed" condition - select the color of the selected ROIs outline, when the ROI Tolerance conditional statement fails.

Composite ROIs selected segment - select the color of the selected segment in a composite ROI.

User Mask ROI - Graphics details - select the display for a User Mask ROI: display only the Bounding rectangle, or display the bounding rectangle and the mask area(s).

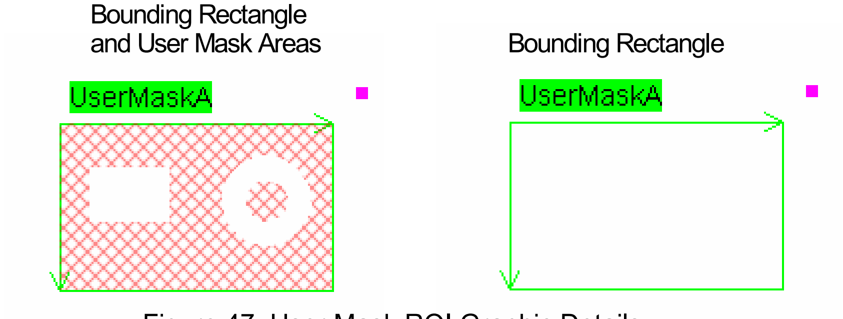


Figure 47. User Mask ROI Graphic Details

Acquisition Options

Open the Acquisition Options from the File menu: Options➔Acquisition.

Configure output IO initial state

Output 0 - Output 7 Set the initial state for each output, when Sherlock opens. The number of outputs changes with the acquisition hardware. You can double-click on the state to toggle it.

Configure camera rotation (degrees)

Image rotation is useful if some cameras must be mounted at different angles to accommodate the space available in the manufacturing environment.

Each camera image can be independently rotated at any angle. The image window size is adjusted to accommodate the size of the rotated image. Rotation is performed by the acquisition driver, before the image is loaded into the Sherlock image buffers. Rotation does not affect Sherlock processing time. Rotation does increase the image acquisition time. Therefore the overall execution time is increased. Rotation at 90-degree increments gives the fastest response and best image quality, because there is no need for interpolation. A flag in the drivers.ini file makes linear interpolation the default method. At 180 degree rotation there is a row by row replacement. At 90 degree rotation there is a row by column replacement, which takes longer.

Configure driver loading

Sapera LT enable or disable loading the Sapera LT driver. Used for Genie series GigE cameras, Sapera supported frame grabbers and CameraLink Vision Appliances. You can double-click on “Enable” or “Disable” to toggle the state.

IFC enable or disable loading the IFC driver “GenICamSetup” used for the PL-USB or the PIC2592 camera. You can double-click on “Enable” or “Disable” to toggle the state.

Video for Windows enable or disable loading the Video for Windows driver. Used for USB and DirectShow cameras. You can double-click on “Enable” or “Disable” to toggle the state.

Sample driver an example of writing a 3rd party driver. You can double-click on “Enable” or “Disable” to toggle the state.

Simulation driver enable or disable loading the software camera simulator, used with a license with no hardware. You can double-click on “Enable” or “Disable” to toggle the state.

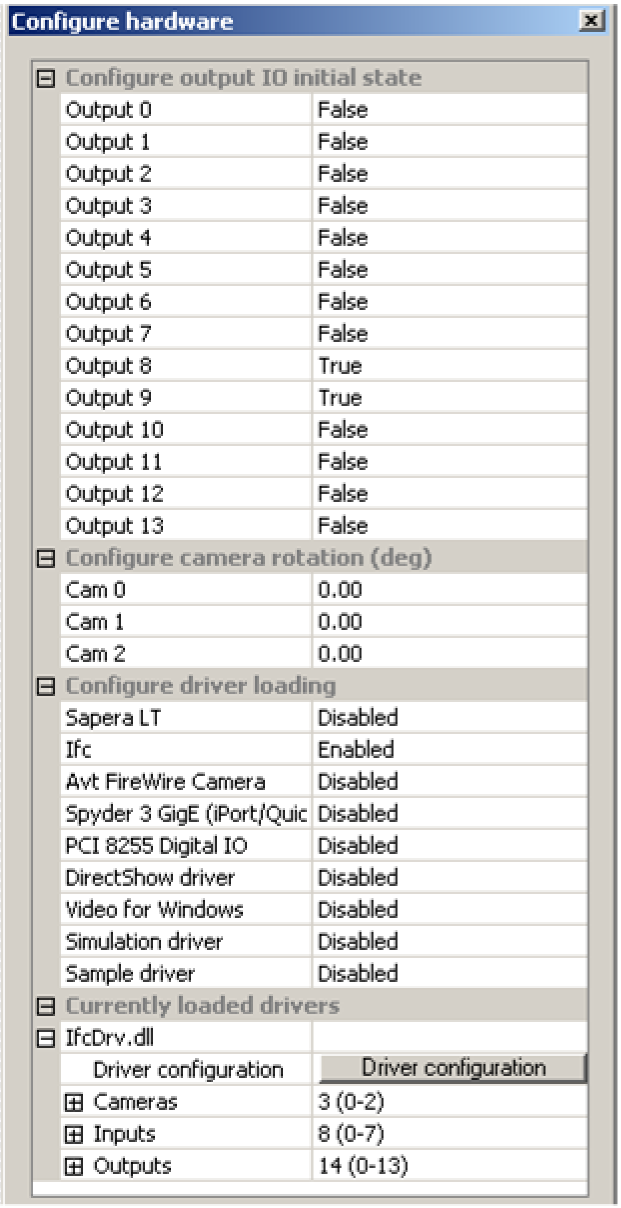
Currently loaded drivers:

Driver configuration - click to open the driver initialization file “*.ini” in notepad. You can view or edit the ini file. Save any changes you make before closing notepad. Changes are not applied until you close, and re-open Sherlock, when the acquisition drivers are loaded.

Cameras - expandable list shows the configured cameras and modes, if supported.

Inputs - expandable list shows the input channels and descriptions.

Outputs - expandable list shows the output channels and descriptions.



Configure hardware

⊟ **Configure output IO initial state**	
Output 0	False
Output 1	False
Output 2	False
Output 3	False
Output 4	False
Output 5	False
Output 6	False
Output 7	False
Output 8	True
Output 9	True
Output 10	False
Output 11	False
Output 12	False
Output 13	False
⊟ **Configure camera rotation (deg)**	
Cam 0	0.00
Cam 1	0.00
Cam 2	0.00
⊟ **Configure driver loading**	
Sapera LT	Disabled
Ifc	Enabled
Avt FireWire Camera	Disabled
Spyder 3 GigE (iPort/Quic	Disabled
PCI 8255 Digital IO	Disabled
DirectShow driver	Disabled
Video for Windows	Disabled
Simulation driver	Disabled
Sample driver	Disabled
⊟ **Currently loaded drivers**	
⊟ IfcDrv.dll	
Driver configuration	Driver configuration
⊞ Cameras	3 (0-2)
⊞ Inputs	8 (0-7)
⊞ Outputs	14 (0-13)

Figure 48. Acquisition Hardware Options

Input/Output Options

Open the Input/Output options from the Options menu: Options➔IO. I/O Options has six subheads: Digital Output, Digital Input, Serial, Tcp/Ip, Modbus, Control Logix, Program Switch.

Note: The Windows Firewall should be turned off to communicate with other devices on the Network.

- IO: TCP/IP instructions, IO: Modbus instructions, and IO: ControlLogix instructions require a previously configured connection.
- IO: Ethernet/IP instructions do not require a previously configured connection. The connection is opened when the first Ethernet IP instruction is executed. Standard ports are used as follows:

a. Communicating with a SLC5 or a MicroLogix controller uses PCCC encapsulation, UDP and TCP ports 0xAF12 (44818).

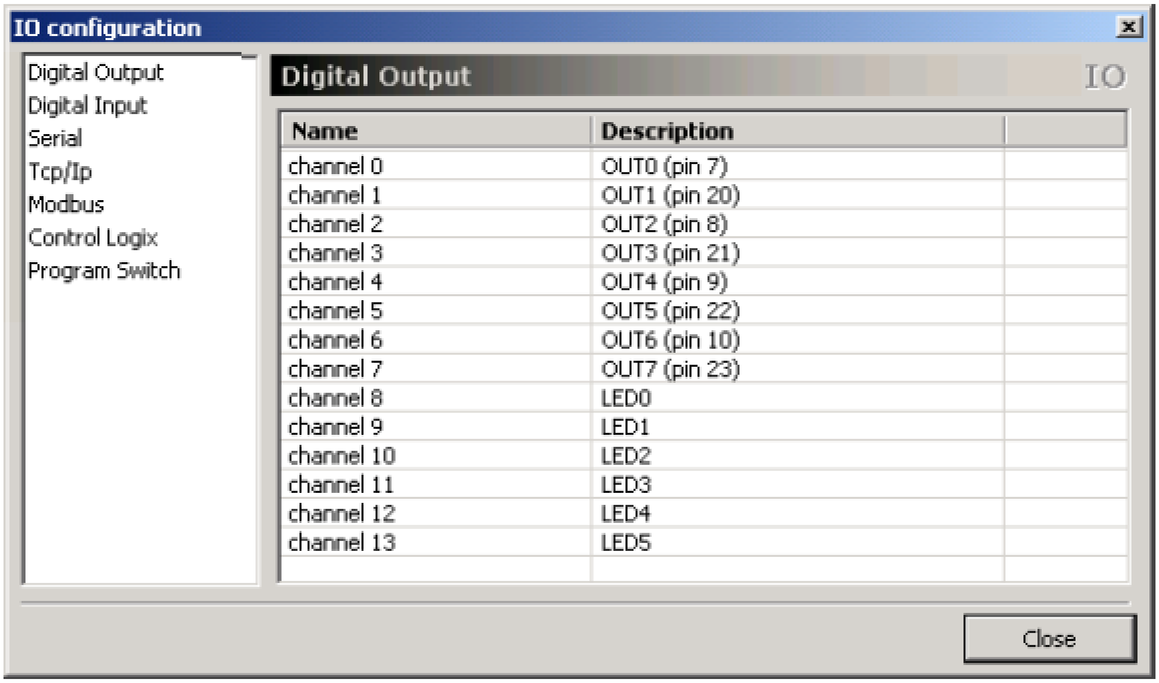
b. Communicating with a ControlLogix controller via Implicit messaging, UDP port 0x8AE (2222).

Digital Inputs and Outputs

Digital Inputs, and Digital Outputs display the available inputs and outputs. **There are no options that you can change in the Digital Input or Digital Output window.**

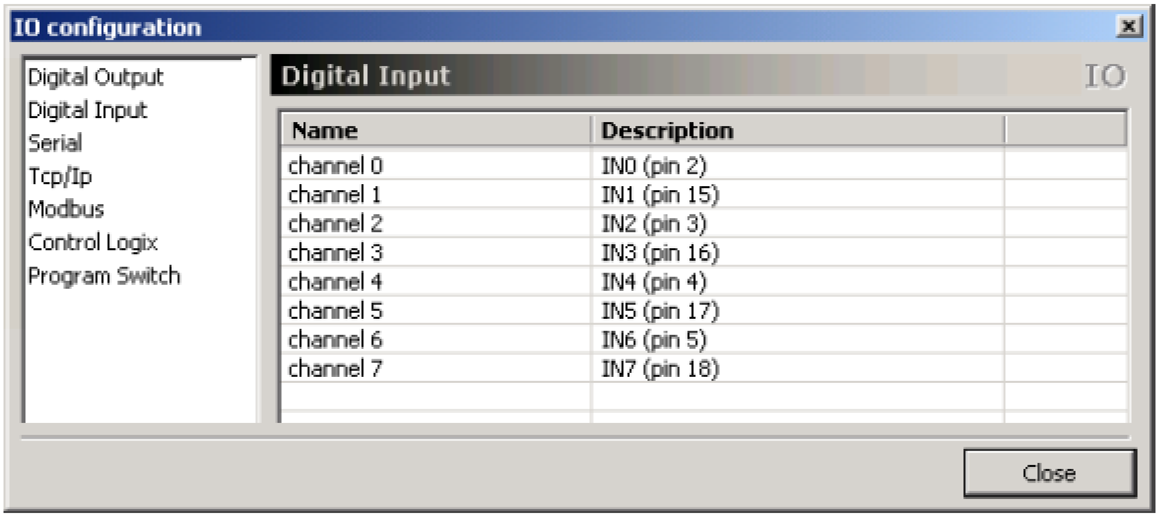
You can set the initial state of the Digital Outputs in the Hardware Options or Acquisition Options dialog (page 65). You can also modify the state of the Digital Outputs in the Digital Outputs window (page 16).

Camera registers may also appear, and may change the channel numbering. The order of multiple devices is determined by the order of the drivers in the Drivers.ini file found in the Sherlock\Drivers directory.



Name	Description
channel 0	OUT0 (pin 7)
channel 1	OUT1 (pin 20)
channel 2	OUT2 (pin 8)
channel 3	OUT3 (pin 21)
channel 4	OUT4 (pin 9)
channel 5	OUT5 (pin 22)
channel 6	OUT6 (pin 10)
channel 7	OUT7 (pin 23)
channel 8	LED0
channel 9	LED1
channel 10	LED2
channel 11	LED3
channel 12	LED4
channel 13	LED5

Figure 49. Digital Outputs Configuration



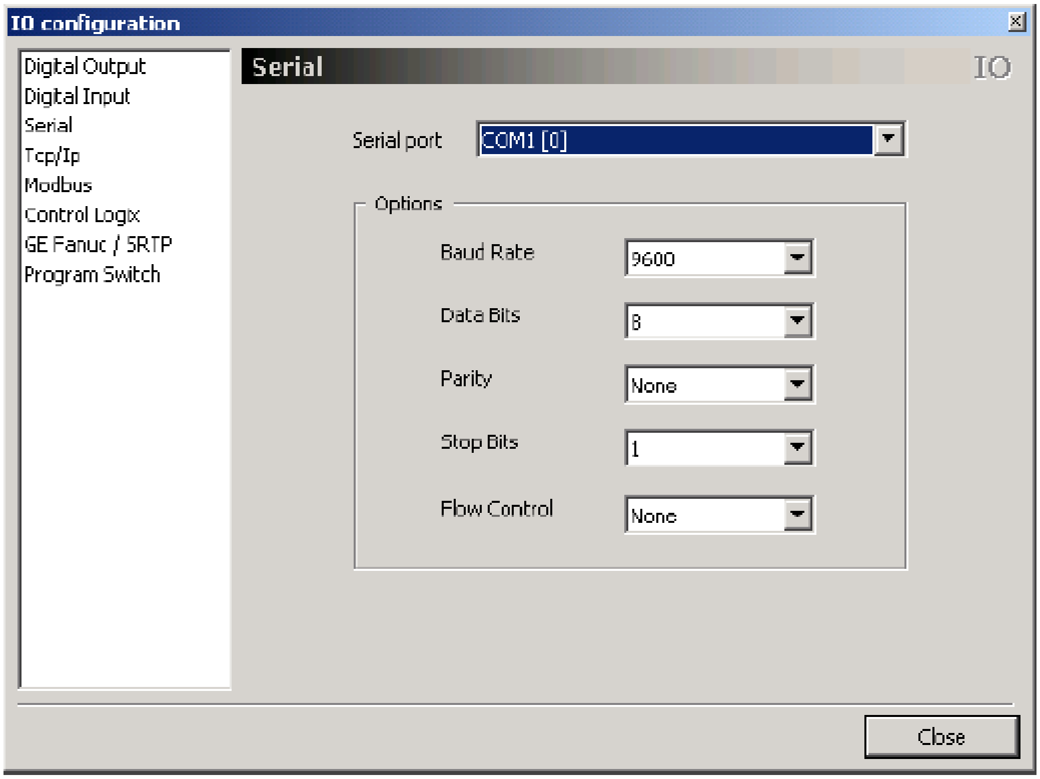
Name	Description
channel 0	IN0 (pin 2)
channel 1	IN1 (pin 15)
channel 2	IN2 (pin 3)
channel 3	IN3 (pin 16)
channel 4	IN4 (pin 4)
channel 5	IN5 (pin 17)
channel 6	IN6 (pin 5)
channel 7	IN7 (pin 18)

Figure 50. Digital Inputs Configuration

Serial Connection

Serial Options allow you to configure the settings of the Serial Ports available on your PC. The Serial port must be configured before it can be used with I/O Control Operations.

Select the serial port and set communication options. Click “Close” to accept your changes.



Serial port: COM1 [0]

Baud Rate: 9600

Data Bits: 8

Parity: None

Stop Bits: 1

Flow Control: None

Figure 51. Serial Configuration

TCP/IP Connection

The TCP/IP Connection must be configured and enabled before it can be used with I/O Control Operations.

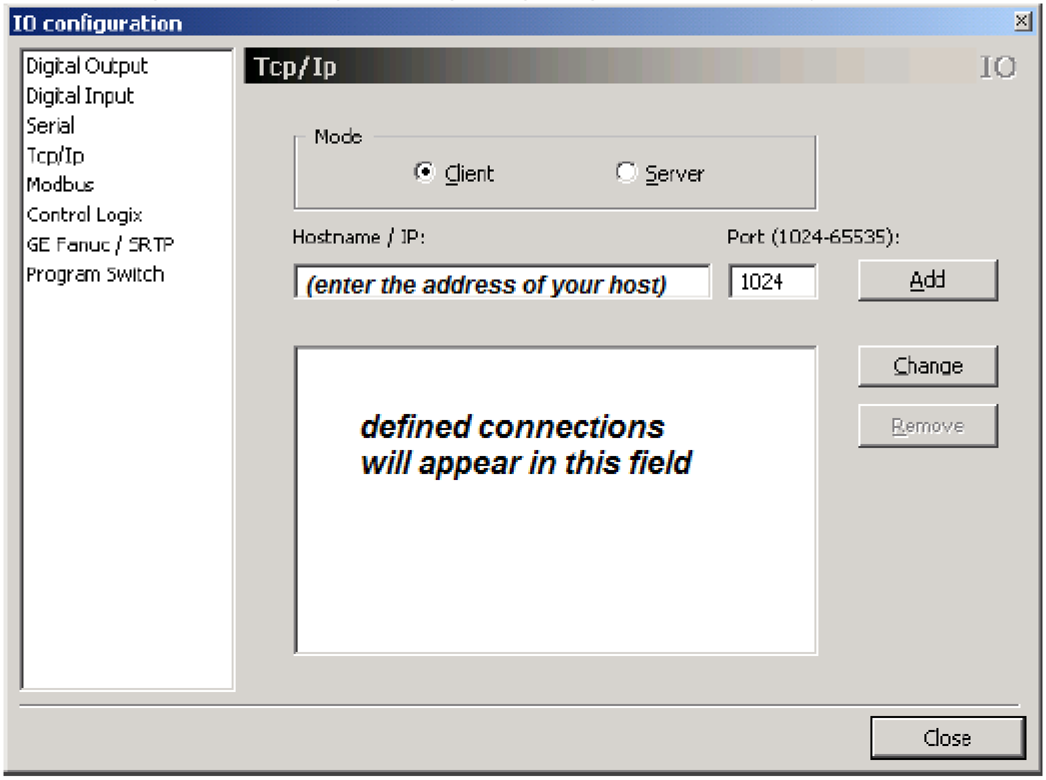
Note: The Windows Firewall should be turned off to communicate with other devices on the Network.

Define a Connection

1. Select **Server** or **Client** mode, enter an IP Address, and click "Add" to add a connection.
2. If Sherlock is a **Client**, enter you host/server's IP Address in the "Hostname/IP" field, and change the "Port" number to match your server.
3. If Sherlock is a **Server**, your client will need the IP Address of this system or PC running Sherlock, and the Port number shown below.
4. Click "Add" to accept your changes.

Change a Connection

1. To change an existing program's connection, enter a new IP Address, or select a defined connection in the list, and Click "Change". Use change for migrating programs to a new system or server.



Mode: ● Client ○ Server

Hostname / IP: (enter the address of your host)

Port (1024-65535): 1024

defined connections will appear in this field

Figure 52. TCP/IP Configuration

Mode - selects Server mode or Client mode. In ***Server*** mode, a client will connect to Sherlock using the assigned ***IP Port*** number entered here. In ***Client*** mode, Sherlock must connect to a Server, using the Server's ***IP Address*** and ***Port*** number. You must enter the Server's correct address and port.

IP Address - the address of a Server that Sherlock connects to in ***Client*** mode.

IP Port - If the mode is *Server*, this is the port number clients must use to connect to Sherlock. If the mode is *Client*, this is the port number of a server that Sherlock will connect to.

Add - click to add your defined connection to the large text field.

Change - click to change the IP Address for the highlighted connection definition. This allows you to change the IP Address without invalidating the handles and instructions in the program.

Remove - select a defined connection in the list, and click to delete that definition.

Modbus Connection

The Modbus Connection must be configured and enabled before it can be used with I/O Control Operations.

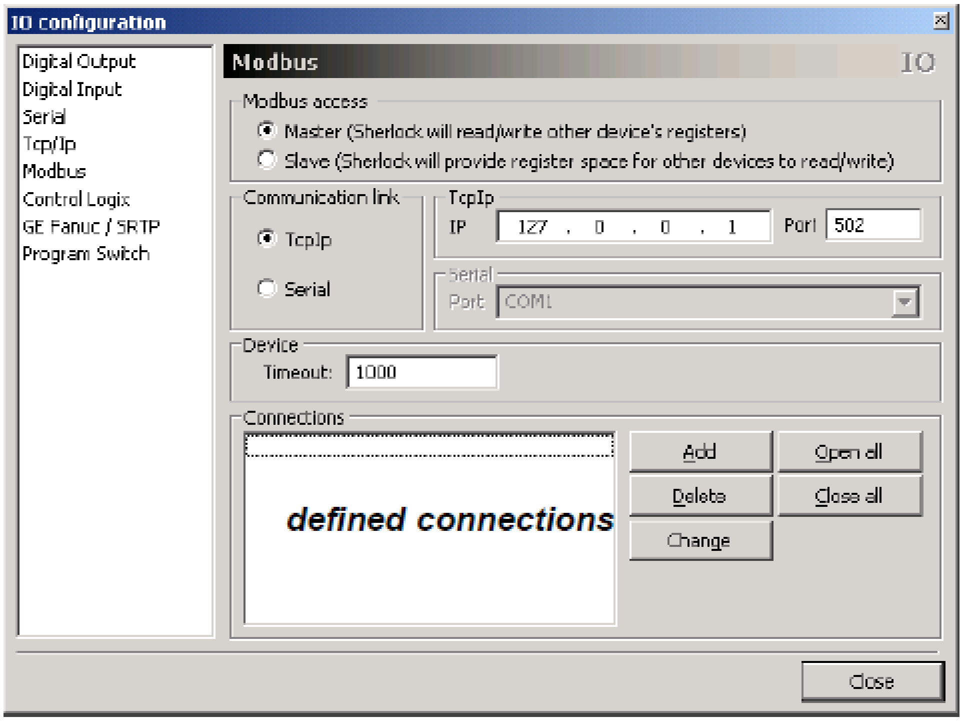
Note: The Windows Firewall should be turned off to communicate with other devices on the Network.

Define a Connection

1. Select Master or Slave for the PC running Sherlock.
2. Select TCP/IP or Serial Port. Select a COM port on the current PC or Vision Appliance.
3. Enter the IP Address and Port number of the Slave, if this is the master.
4. Click Add ot accept your changes.

Changing the Connection IP

1. Enter a new IP Address and Click Change, or select a defined connection and click Change.



IO configuration

Modbus

Modbus access

● Master (Sherlock will read/write other device's registers)

○ Slave (Sherlock will provide register space for other devices to read/write)

Communication link

● TcpIp

○ Serial

TcpIp

IP 127 . 0 . 0 . 1 Port 502

Serial

Port COM1

Device

Timeout: 1000

Connections

Figure 53. Modbus Configuration

Modbus access - select master or slave operation.

- *Master:* Sherlock will read and write to another devices registers.
- *Slave:* Sherlock will provide register space for another device to read and write to.

Communication link - selects a Serial or TCP/IP connection, and enables parameter fields under TCP/IP or Serial for your modification.

Timeout - set a time out period for Master read/write access.

Add - click to add your defined connection (defined connections appear in the large text field).

Delete - click to remove a highlighted definition from the large text field.

Change - click to change the IP Address for the highlighted connection definition. This allows you to change the IP Address without invalidating the handles and instructions in the program.

Open all - open all defined Modbus connections. Test/verify definitions and availability.

Close all - close all defined Modbus connections.

Control Logix Connection

The Control Logix connection must be configured and enabled before it can be used with I/O Control operations. You also must select which variables will be available as handles in I/O Control instructions. Refer to the step-by-step instructions on page 72.

Note: The Windows Firewall should be turned off to communicate with other devices on the Network.

- Communicating with a Control Logix controller via Explicit messaging, TCP port 0xAF12 (44818).

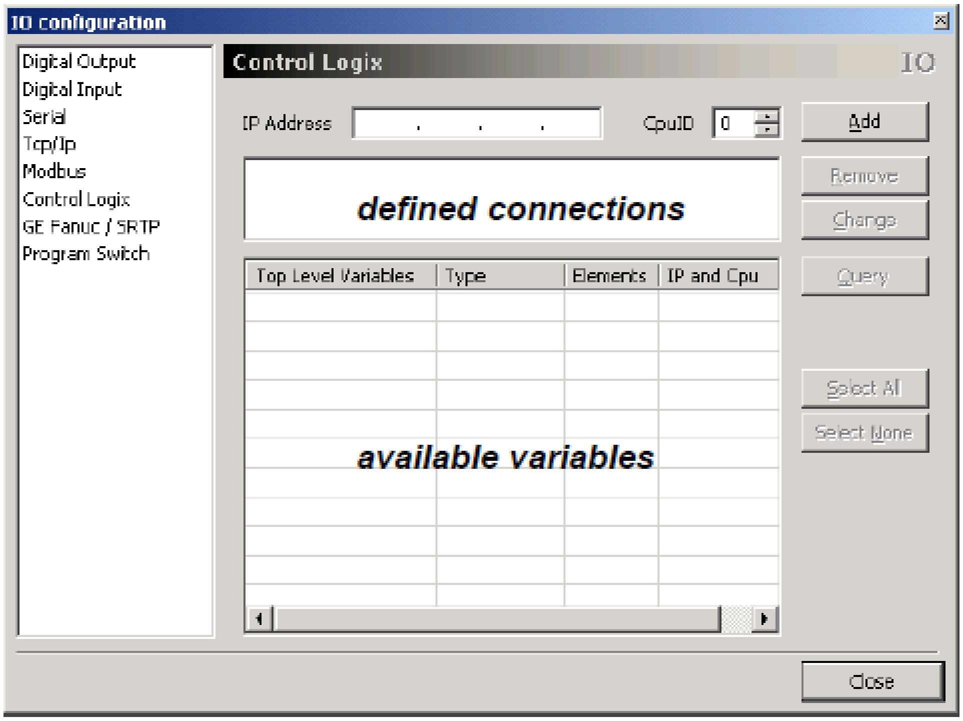


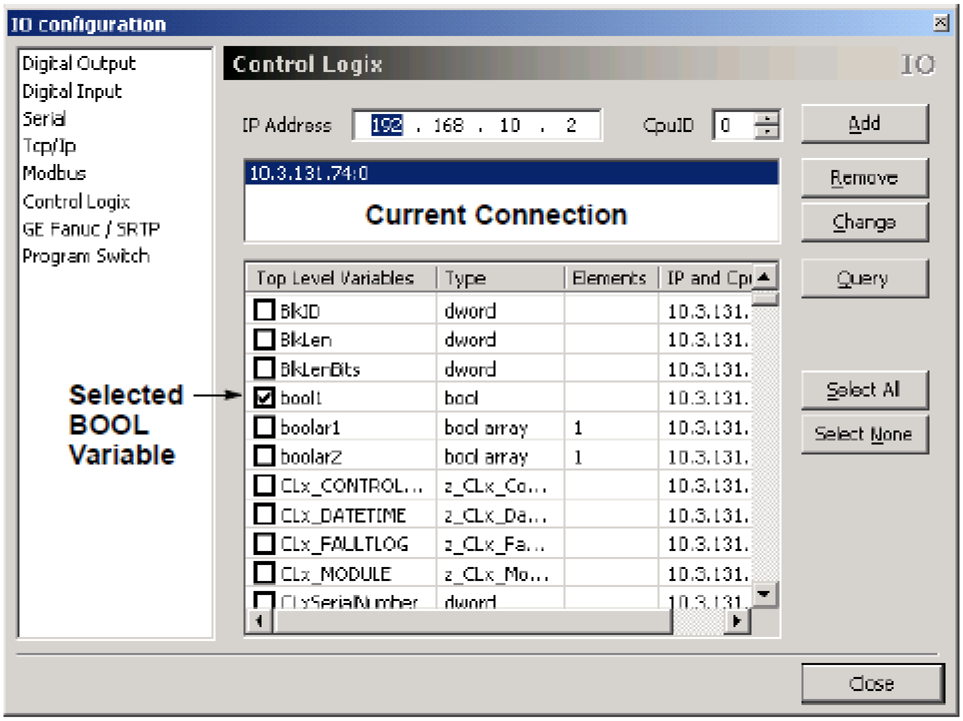
Figure 54. Control Logix Configuration

Defining a Connection

1. Enter the IP Address and CPU ID (usually 0) of your ControlLogix controller, then click “Add”. The IP Address, with the CPU ID appended, will appear in the middle field (figure 55). A connection is uniquely identified by the pair <IP address>:<CPU ID>. The connection is kept open as long as it is present in the middle field. *Use only valid addresses. It will take a while to time out every time Sherlock attempts to re-open an invalid connection.*

2. Click on the controller address in the middle field, then click “Query” to interrogate the controller for variables. The list of variables will appear in the bottom field.

3. Use the check boxes in the list of variables (bottom field), to select which variables will be available as handles in the IO instructions. You can use the “Select All” and “Select None” buttons to select and de-select variables. See Figure 55.



IP Address 192 . 168 . 10 . 2 CpuID 0

10.3.131.74:0

Top Level Variables	Type	Elements	IP and Cpu
□ BlkID	dword		10.3.131.
□ BlkLen	dword		10.3.131.
□ BlkLenBits	dword		10.3.131.
☑ bool1	bool		10.3.131.
□ boolar1	bool array	1	10.3.131.
□ boolar2	bool array	1	10.3.131.
□ CLx_CONTROL...	z_CLx_Co...		10.3.131.
□ CLx_DATETIME	z_CLx_Da...		10.3.131.
□ CLx_FAULTLOG	z_CLx_Fa...		10.3.131.
□ CLx_MODULE	z_CLx_Mo...		10.3.131.
□ ClxSerialNumber	dword		10.3.131.

Figure 55. Selecting ControlLogix Variables

Changing the Connection IP

Enter a New IP Address and Click "Change". This changes the Controller address without invalidating the handles in the program. The new controller is not queried automatically. The handles and variables are updated with the new IP Address. Variable selections are not changed in the list.

IP Address - the address of the ControlLogix controller.

CPU ID - a sequential index (usually zero). Supports more than one ControlLogix CPU at the same IP address.

Add - click to add your defined connection. Defined connections will appear in the middle field in this menu. See figure 55.

Remove - select a controller in the middle field, then click to delete that connection. You will be warned if any variables associated with that controller are used in instructions.

Change - click to change the IP Address for the highlighted connection definition. This allows you to change the IP Address without invalidating the handles and instructions in the program.

Query - select a controller in the middle field, then click to query the controller for available variables or registers.

Select All - select all variables in the list (bottom field).

Select None - un-select all variables in the list (bottom field).

GE Fanuc Connection

The GE Fanuc SRTP connection must be configured and enabled before it can be used with I/O Control operations.

Note: The Windows Firewall should be turned off to communicate with other devices on the Network.

Defining a Connection

1. Enter the IP Address of the Controller (PLC) and click "Add". The address will appear in the middle field.

NOTE *Use only valid addresses. It will take a while to time out every time Sherlock attempts to re-open an invalid connection.*

Changing the Connection IP

1. To change the IP Address for a program that already contains a GE Fanuc definition, enter the new IP Address or select a connection from the list, and click "Change".

This changes teh selected connection but does not invalidate handles in a program. This is the preferred method when changing to a new controller or migrating a program to a new system.

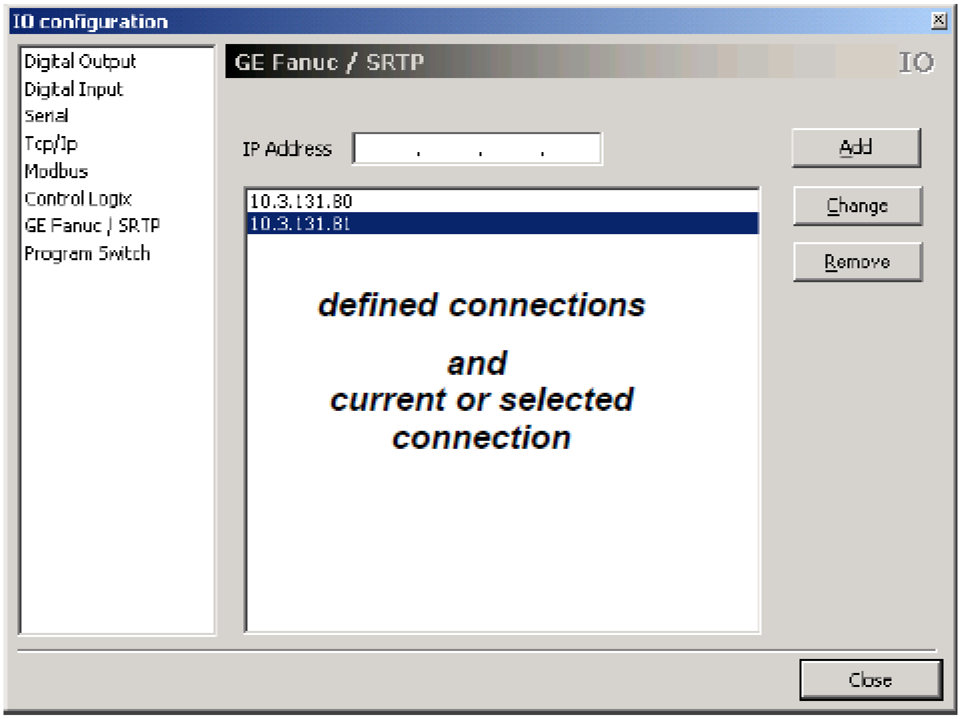


Figure 56. GE Fanuc Configuration

IP Address - the address of the GE Fanuc controller.

Add - click to add your defined connection. Defined connections will appear in the middle field in this menu.

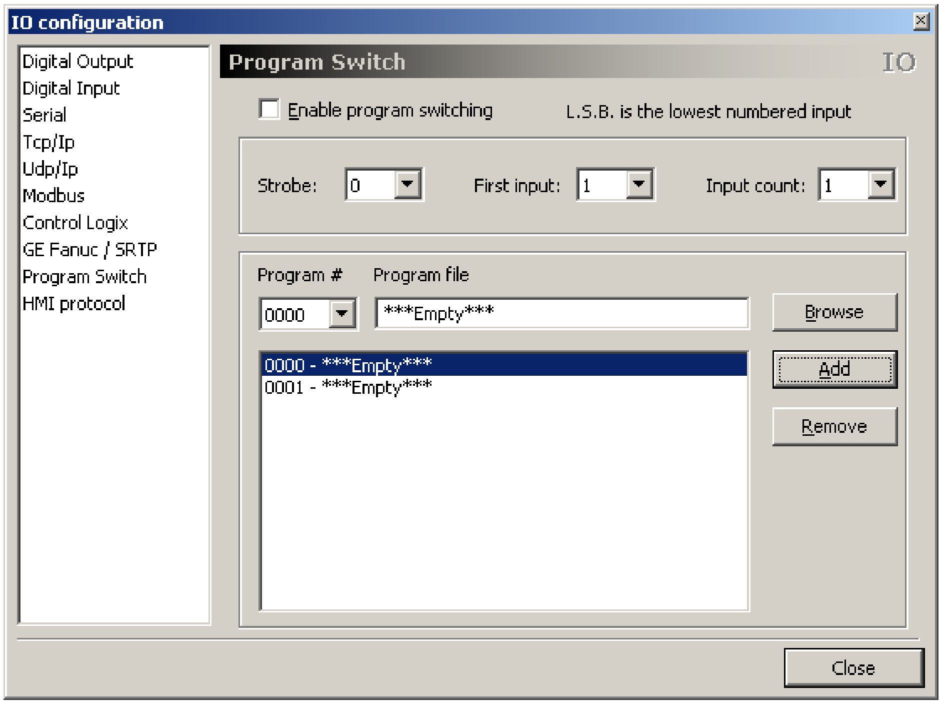
NOTE *Notice there are two connection in the figure. The highlighted connection is the selected or current connection used by instructions in the program.*

Change - click to change the IP Address for the highlighted connection definition. This allows you to change the IP Address without invalidating the handles and instructions in the program.

Remove - select a controller in the middle field, then click to delete that connection. You will be warned if any variables associated with that controller are used in instructions.

Program Switch Configuration

Program switching uses the digital input lines to select a program (investigation file). The "Strobe" input halts the running program, loads the selected investigation and runs. You must define the number of digital inputs, and assign investigation file names to numbers.



□ Enable program switching

L.S.B. is the lowest numbered input

Strobe: 0 First input: 1 Input count: 1

Program # 0000 Program file ***Empty***

0000 - ***Empty***
0001 - ***Empty***

Figure 57. Program Switch Configuration

Enable program switching - enables the rest of this window, and enables switching investigation files based on the Digital Inputs.

Strobe - you must assign one digital input as the "Enable" or "Switch Program" command input. In operation, a binary number is written to the input pins then the Strobe is enabled to load the number and affect the program switch (load a investigation file).

First input - defines which digital input is the Least Significant Bit of the binary number code. You must use adjacent of sequential bits for the binary number code. There is no method to specify discontinuous bits for this assignment.

Input count - defines how many adjacent or sequential bits to use for forming a binary number. The number of programs that can be assigned and switched, is dependent on the number of bits used. i.e. 3 bits are necessary to assign 8 programs (0 through 7). *Sherlock cannot detect conflicts between this configuration window and other input assignments in a investigation or program.*

Program - drop list shows the binary numbers available, as set by the Input Count.

Program file - enter a full path to the investigation file, or use the "Browse" button to locate a file. Click "Add" to add the assignment of number and file to the list in the lower pane.

Add - add a number and program file assignment to the list in the bottom pane.

Remove - remove the selected or highlighted assignment from the list in the bottom pane.

NOTE *Do not leave "***Empty***" assignments. You should create a dummy ivs file program that would indicate to the user that no program matches the number selection, and stop Sherlock. Sherlock will not detect an invalid or Empty assignment. No new program is loaded, and the old program will continue to run.*

If a program fails to load (for example there is an assignment, but the program has been deleted or renamed) the previous investigation is saved and a New (empty) investigation program is created and Sherlock stops running. Sever program load errors may halt Sherlock without saving the previous investigation.

If a program loads successfully, the program starts to run in "Run Continuous" mode.

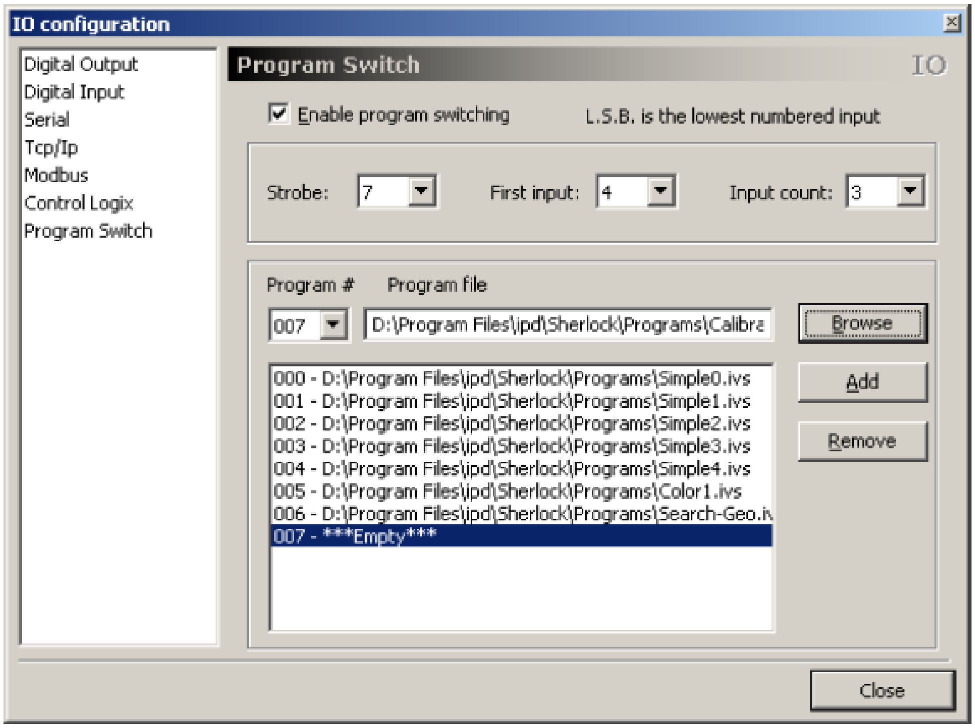
The settings for this feature are saved in the "IpePgmSw.dat" file created in the \Sherlock\bin directory. This file is updated at the same time the rest of the Sherlock application settings are saved or loaded.

If you switch to the current running program, Sherlock re-loads, initializes and begins running. Sherlock does not ignore a request to switch to the current program.

Configuring Program Switching

1. Check the box beside "Enable program switching" to enable editing the other fields.
2. Select the strobe bit, LSB (First input) and number of bits used (Input count).
   - the input bits must be contiguous (no gaps between)
   - the strobe must be outside the range of inputs
3. Use the pull down or click on a number assignment in the list of assignments field. Use the "Browse" button to find the program investigation files.
4. Click the "Add" button to add an assignment to the list. Assignments are not added until you click "Add".

Figure 58 shows the strobe assigned to bit 7 (Input Chanel 7) and bits 4 through 6 (Input Channels 4, 5, 6) assigned as inputs to form the binary program number.



☑ Enable program switching   L.S.B. is the lowest numbered input

Strobe: 7   First input: 4   Input count: 3

Program #: 007   Program file: D:\Program Files\ipd\Sherlock\Programs\Calibra

```
000 - D:\Program Files\ipd\Sherlock\Programs\Simple0.ivs
001 - D:\Program Files\ipd\Sherlock\Programs\Simple1.ivs
002 - D:\Program Files\ipd\Sherlock\Programs\Simple2.ivs
003 - D:\Program Files\ipd\Sherlock\Programs\Simple3.ivs
004 - D:\Program Files\ipd\Sherlock\Programs\Simple4.ivs
005 - D:\Program Files\ipd\Sherlock\Programs\Color1.ivs
006 - D:\Program Files\ipd\Sherlock\Programs\Search-Geo.i
007 - ***Empty***
```

Figure 58. Assigning Investigation Files

NOTE *Notice in Figure 58 the "Program #" field shows "007" and the "Program file" field shows a path to a valid investigation file, but the assignments list shows "***empty***" beside "007".* ***The assignment is not made until you click on "Add".***

HMI Connection

Sherlock can communicate with an HMI Panel that is running the BOAView software. This panel allows you to define the IP Address and Port number for output to the HMI.

Note: The Windows Firewall should be turned off to communicate with other devices on the Network.

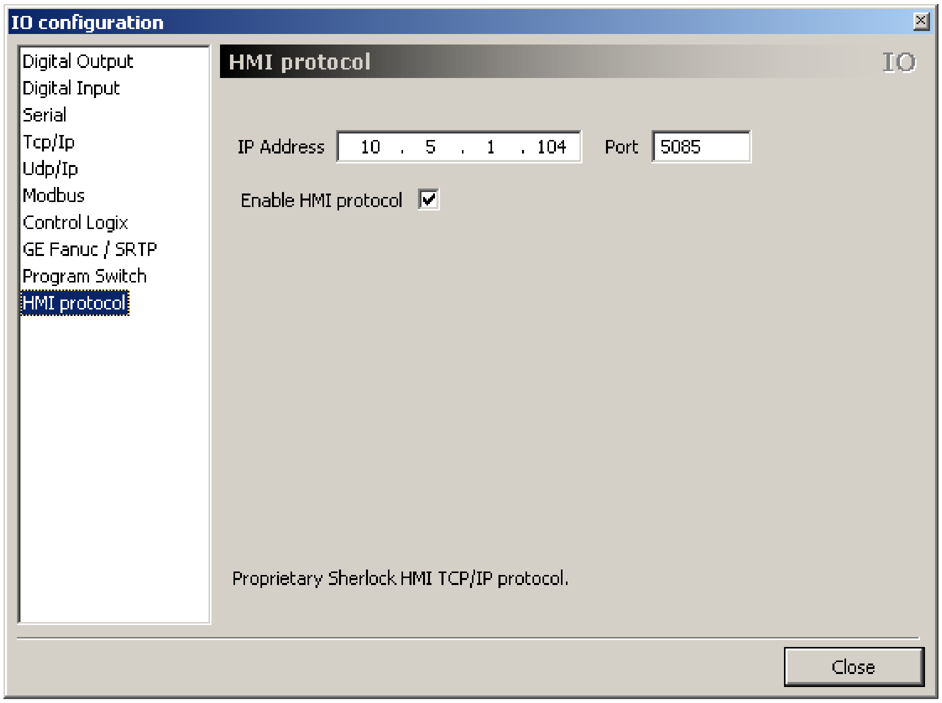
1. Enter the **IP Address** of the NIC (network interface card) ***Sherlock*** uses to ***output*** images and data for HMI. On a PC, this can be any one of multiple NIC or LAN ports available. **NOTE:** This is **not** the IP of the HMI panel.

2. Enter 5085 as the **Port Number**. This is the dedicated port number for the "BOAView" software running on the HMI device.

3. Check the box beside **Enable HMI Protocol**.

4. Click **Close**. Save the investigation file.

The HMI Protocol settings are saved in the Investigation file (*.ivs or *.ivx). You must configure and enable the HMI Protocol for each new investigation file you create.



IP Address 10 . 5 . 1 . 104 Port 5085

Enable HMI protocol ☑

Proprietary Sherlock HMI TCP/IP protocol.

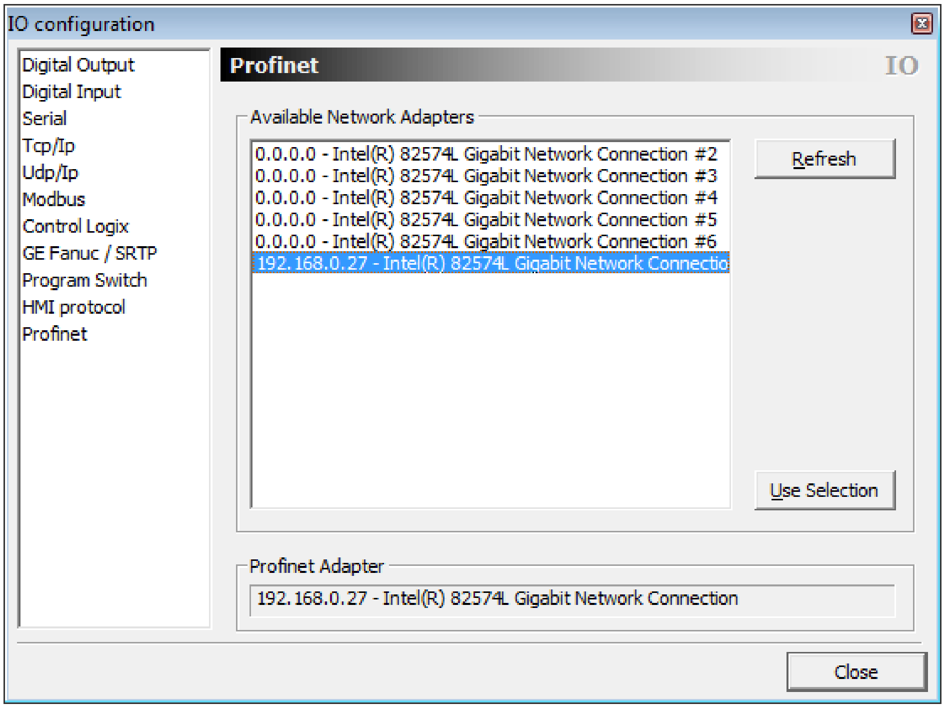
Figure 59. HMI Address Configuration

Profinet Connection

Sherlock can communicate with a PLC using the Profinet protocol.

Note: VICORE automatically selects the dedicated PLC Port for Profinet communication. You do not need to follow the steps below on the VICORE. Make sure your VICORE PLC Port is configured for Profinet.

1. Select the port you will use for Profinet communication,
2. Click "Use Selection" to populate the "Profinet Adapter" field below.



Available Network Adapters

0.0.0.0 - Intel(R) 82574L Gigabit Network Connection #2
0.0.0.0 - Intel(R) 82574L Gigabit Network Connection #3
0.0.0.0 - Intel(R) 82574L Gigabit Network Connection #4
0.0.0.0 - Intel(R) 82574L Gigabit Network Connection #5
0.0.0.0 - Intel(R) 82574L Gigabit Network Connection #6
192.168.0.27 - Intel(R) 82574L Gigabit Network Connectio

Profinet Adapter: 192.168.0.27 - Intel(R) 82574L Gigabit Network Connection

Figure 60. Profinet IO Configuration

In this illustration all the other ports are disconnected. That is why they all show no IP Address. This makes it very easy to spot the correct port. If you have cameras connected, their addresses will appear in this list.

NOTE: There are several steps to configuring the NIC Port used for Profinet communication. Please refer to the Sherlock 7 Communication Guide.

JavaScripts

Sherlock supports JavaScript inserted directly into the investigation program, or running in background. To insert a javascript in your program:

- Click on the instruction before your script insertion, and click on the “Script” button in the Program window toolbar,
- Drag the “Script code” instruction from the Instruction window into the Program window.

Double-left-click on the “scriptA” instruction in the Program window to open the JavaScript editor. There are three panes in the Script editor window: Code, Variables, and Predefined program items. The code window is enabled to Auto-Complete on some of the predefined items or objects.

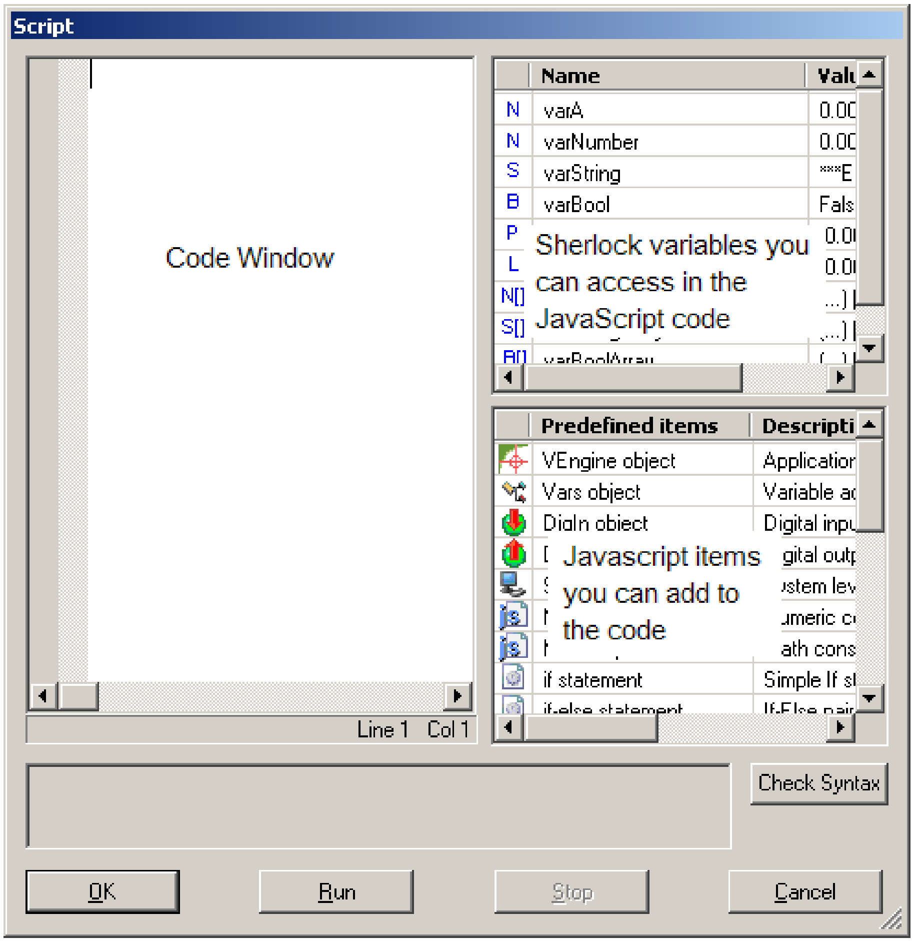


Figure 62. Editing Scripts

Variables - the variables pane lists all the variables defined in your Sherlock investigation. Sherlock variables are contained in JavaScript variable **Vars** object. If you drag a variable “varA” into the code pane, it looks like: “Vars.varA”. If you are typing in the code pane, and type **Vars.** when you type the period, a window pops up with the available variables.

Check Syntax - check standard JavaScript syntax rules.

Run - executes the script in the editor window, in the context of the stopped Sherlock program. A thread is manually started, and the script is executed in that special thread. The rest of the Sherlock program does not execute at this time. This may be useful for checking generic code snippets for run-time errors in syntax or logic, but it will not work for all possible valid scripts because the Sherlock program is halted. Make sure you understand the implications of the concrete code before attempting to run a particular script. **NOTE:** There is no indication that a script is running unless you write to the monitor or the reporter in your script. Variables changed during the script execution are not updated in real time. Once you close the editor, you must force a refresh of the Watch or Variable windows so they will reflect the changes in the variable values.

Stop - halt the running script. If a script takes too long to finish (the OK, Run & Cancel buttons are not enabled) you can terminate it. If the script is hung on some call outside the JavaScript interpreter, the only way to stop it may be to kill the Sherlock process, which results in losing all changes.

Online JavaScript Reference

To open the *Javascript* online *reference*, Click on the Windows Start button:
Start ➔ Programs ➔ Teledyne DALSA ➔ Sherlock ➔ JavaScript Help.

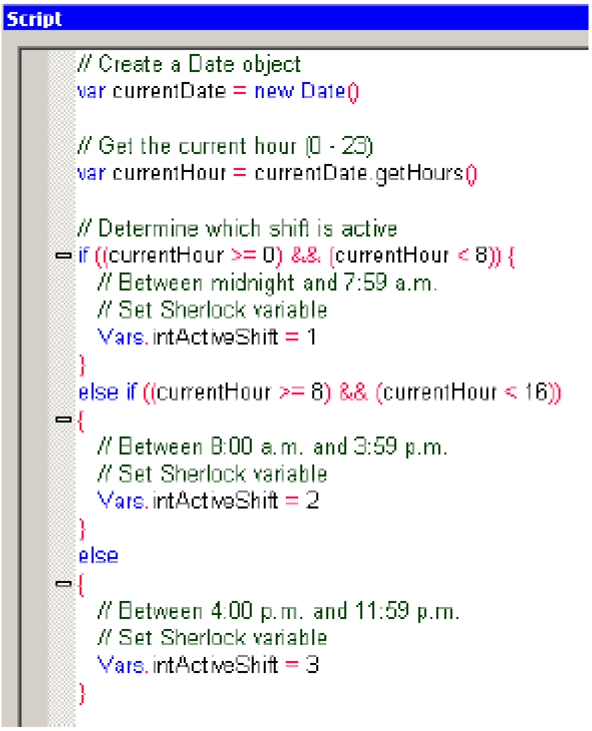
Entering JavaScript

Predefined items - The JavaScript language includes many common programming constructs: **For**, **While**, **Do/While**, **If**, **If./Else**, **Switch**. There are also several objects with built-in properties and methods: **Math**, **Number**, **Vars**, **System**, **VEngine**, etc.

To add code to the code window, you can either type it, or drag-and-drop an item from the *Predefined Items* or Sherlock ***variables*** pane. When you add an object from *Predefined Items* and then type a period (.), the object's properties and methods are displayed in a drop list.

Sherlock variables are accessed from the **Vars** object. If you type **Vars.** (with the period) in the code window, the variables are displayed in a drop-down list. If you drag-and-drop **Vars** from the **Predefined items** list to the code window, then type a period, the same drop-down list is displayed. If you drag-and-drop a variable from the variable window, **Vars.** is automatically pre-pended.

This example of JavaScript code gets the current hour from the **Date()** object, and determines which 8-hour shift is active. The Sherlock variable **intActiveShift** is set to 1, 2 or 3, accordingly.



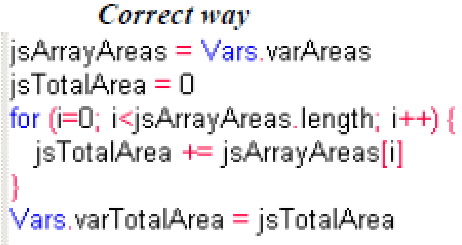
```
// Create a Date object
var currentDate = new Date()

// Get the current hour (0 - 23)
var currentHour = currentDate.getHours()

// Determine which shift is active
if ((currentHour >= 0) && (currentHour < 8)) {
  // Between midnight and 7:59 a.m.
  // Set Sherlock variable
  Vars.intActiveShift = 1
}
else if ((currentHour >= 8) && (currentHour < 16))
{
  // Between 8:00 a.m. and 3:59 p.m.
  // Set Sherlock variable
  Vars.intActiveShift = 2
}
else
{
  // Between 4:00 p.m. and 11:59 p.m.
  // Set Sherlock variable
  Vars.intActiveShift = 3
}
```

A Sherlock point variable is accessed through two properties, .x and .y. To read the x and y coordinates of a single Sherlock point variable **Point** in JavaScript, use **Vars.Point.x** and **Vars.Point.y**. To read the x and y coordinates of one element of a point array variable **Points**, use **Vars.Points[array_index].x** and **Vars.Points[array_index].y**.

If you need to read more than a few elements from a Sherlock array variable, it is more efficient to copy the entire variable to a JavaScript variable than to access the elements of the Sherlock variable one at a time.

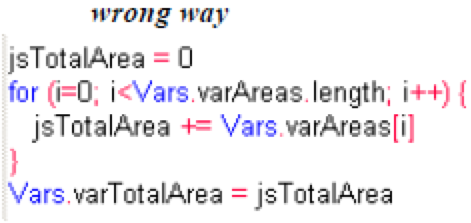
In the following two examples, **Vars.varAreas** is a Sherlock number array variable, and **Vars.varTotalArea** is a Sherlock number variable.

Correct way



```
jsArrayAreas = Vars.varAreas
jsTotalArea = 0
for (i=0; i<jsArrayAreas.length; i++) {
  jsTotalArea += jsArrayAreas[i]
}
Vars.varTotalArea = jsTotalArea
```

wrong way



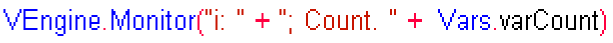
```
jsTotalArea = 0
for (i=0; i<Vars.varAreas.length; i++) {
  jsTotalArea += Vars.varAreas[i]
}
Vars.varTotalArea = jsTotalArea
```

You do not have to define a JavaScript variable before you assign a Sherlock variable to it (for example, **jsArrayAreas = Vars.varAreas**); the JavaScript variable's data type and dimensions (if the Sherlock variable is an array) are automatically set by the assignment.

NOTE *Variables are dynamic in JavaScript. There is no type checking when passing values in Sherlock variables. In JavaScript. 1 and 0 are numbers, TRUE and FALSE are booleans. This can cause problems when passing back to Sherlock.*

NOTE *A JavaScript module has access to Sherlock variables only - not readings.*

NOTE *A JavaScript module cannot call a Sherlock subroutine, nor in any other way interact with an investigation (except by reading and writing Sherlock variables, and calling a* ***VEngine.Monitor*** *method).*



```
VEngine.Monitor("i: " + "; Count: " + Vars.varCount)
```

NOTE *A file opened with Sherlock's* ***IO:File*** *instructions cannot be simultaneously accessed by the JavaScript* ***System*** *object operations (****FileClose, FileAppend,*** *etc.), and vice-versa.*

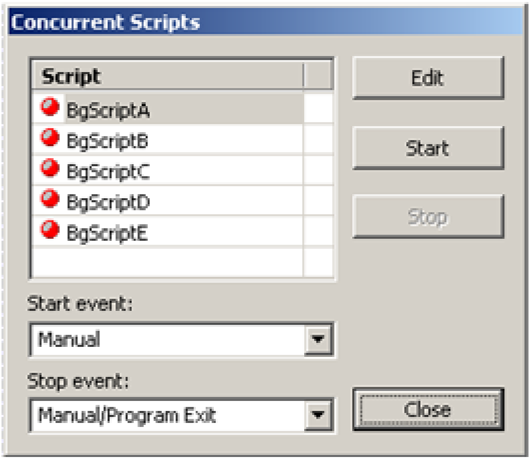
Concurrent Script Options

Open the Concurrent scripts options from the Options menu: Options ➔ Concurrent Script.

The Concurrent Scripts Option window allows you to edit and configure Background Scripts, which run concurrently with your Sherlock Application. Sherlock supports up to five concurrent scripts running in background, independent of Sherlock's investigation.

The Sherlock program and background scripts cannot share access to hardware or system resources. Scripts cannot be synchronized to Sherlock program events.

Background scripts should be written as a loop, and should have a wait or sleep statement. If not written as a loop, the script will execute once and terminate.



Concurrent Scripts

Script
● BgScriptA
● BgScriptB
● BgScriptC
● BgScriptD
● BgScriptE

Start event: Manual

Stop event: Manual/Program Exit

Edit - create or edit the script file and check syntax. Uses the same javascript edit window as scripts inserted in the program window.

Start - starts the *selected* or highlighted script running in background.

Stop - stops running the *selected* or highlighted script.

Start event - select when the *selected* script starts running in background:

- *Manual:* the script is started manually only, using this window.
- *Investigation load:* when Sherlock loads an investigation.
- Continuous Investigation start: when a Continuous investigation is started (not each loop).

Stop event - select when the *selected* script selected is stopped:

- *Manual/Program stop:* manually stopped in this window, or when the Sherlock program is closed.
- *Continuous investigation Stop:* stop when a continuous investigation is halted.

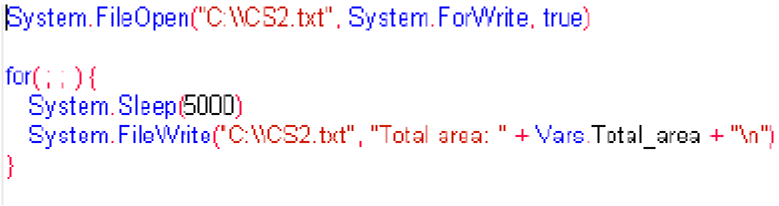
From the **Start event** and **Stop event** drop lists, select the events that will launch and terminate the script. A script is usually launched by the **Investigation Load** or **Continuous Investigation Start** event.

Click the **Edit** button to open the script editor. The editor is exactly the same as for creating JavaScript modules that are called from the main program (Figure 61).

A script is executed only once. To make any or all of the code in a script execute repeatedly, the code must be placed with a **for**, **while**, or **do…while** statement that will loop "forever", until a defined condition is met, or until the selected stop event occurs.

NOTE *For the script to yield control (especially in the case of tight loops) and improve overall application responsiveness, a call to the **Sleep** function call should appear somewhere in the loop.*

The following example script opens the text file **CS2.txt.** Within the **for** loop, the script goes to sleep for 5000 milliseconds; wakes up; and writes a line of text, including the current value of the Sherlock variable **Total_area**, to the file. If the **Sleep** and **FileWrite** statements were not placed within the **for** loop, only one line would be written to the file. Because there is no terminating condition to the **for** loop, it will loop "forever" (until the script is terminated by the stop event).

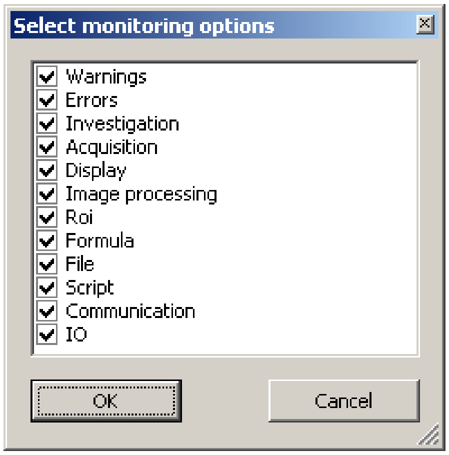


```
System.FileOpen("C:\\CS2.txt", System.ForWrite, true)

for( ; ; ){
  System.Sleep(5000)
  System.FileWrite("C:\\CS2.txt", "Total area: " + Vars.Total_area + "\n")
}
```

NOTE *Different threads of an investigation - the main program and concurrent scripts - cannot access the same resource at the same time. If thread A tries to access a resource that is in use by thread B, thread A will be blocked until thread B releases the resource. Resources include variables, digital IO lines, and files.*

Monitor Options

Click on the "Options" button in the Monitor window toolbar, to open the monitor options. You can select (check) or deselect (uncheck) items to be reported in the monitor window.

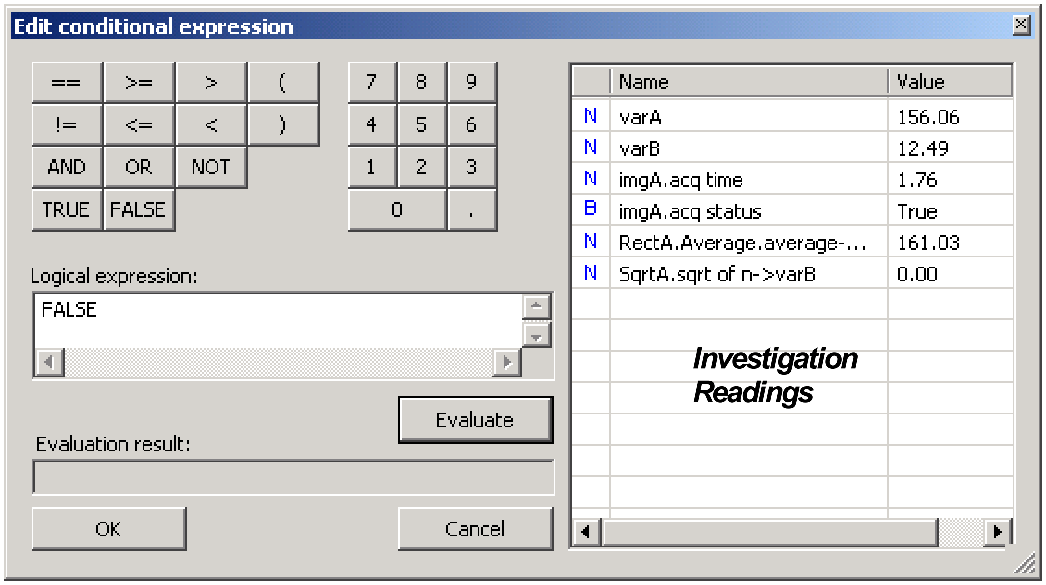


- ☑ Warnings
- ☑ Errors
- ☑ Investigation
- ☑ Acquisition
- ☑ Display
- ☑ Image processing
- ☑ Roi
- ☑ Formula
- ☑ File
- ☑ Script
- ☑ Communication
- ☑ IO

Figure 63. Monitor Options

Conditional Expression

To open the Edit Conditional expression dialog, double-left-click on a "If", "If-Else" or "While" statement in the Program window. The Conditional Expression is where you test limits on Sherlock readings. (In Sherlock 6 you entered tolerance numbers using the Edit Readings dialog.) You can test more than one limit, or more than one reading in one conditional expression, or you can create multiple conditional expressions for different program behaviors. See the Tutorial, page 31.



Logical expression:
FALSE

	Name	Value
N	varA	156.06
N	varB	12.49
N	imgA.acq time	1.76
B	imgA.acq status	True
N	RectA.Average.average-...	161.03
N	SqrtA.sqrt of n->varB	0.00

Investigation Readings

Figure 64. Edit Conditional Expression

Operators and Numbers - Use the buttons to insert operators and numbers in the "logical expression" field. You can also type in the field.

Logical Expression - create an expression by typing, using the operators and numbers buttons, and dragging readings into this field to create a test condition to pass or fail.

Readings - Drag readings into the "logical expression" field. All readings created in the investigation are listed in the right pane. As you create more readings through instructions or algorithms, they will appear in this list.

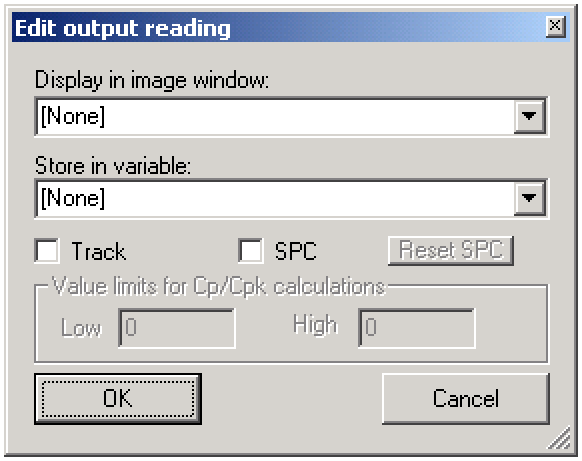
Evaluate - click to see the current result (True or False) displayed in the "Evaluation result" field.

OK - accept changes to the logical expression, and close dialog.

Cancel - discard changes to the logical expression, and close dialog.

Output Reading

Double-left-click on any output reading to open the edit dialog.



Edit output reading

Display in image window: [None]

Store in variable: [None]

☐ Track ☐ SPC

Value limits for Cp/Cpk calculations

Low 0 High 0

Figure 65. Edit Output Reading

Display a reading in an image window - pull down the list to select an image window.

Store reading in a variable - pull down the list to select and assign a variable. (Variables must be created in the Variables window first.)

Track - opens a graph display of a reading value.

SPC - enable SPC data gathering.

Reset SPC - zeros, clears, or resets the data.

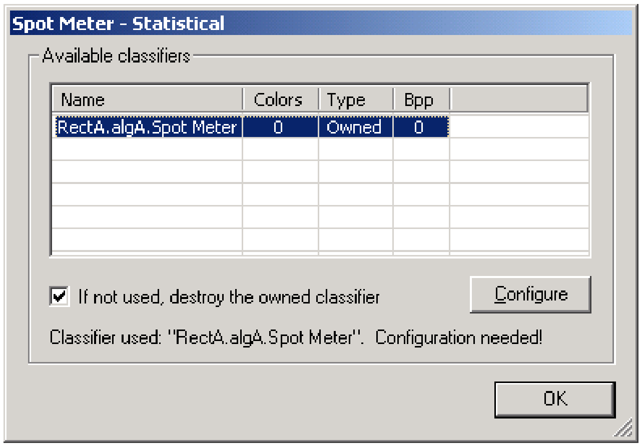
Value limits - assign high and low limits for Cp and Cpk calculations.

Color Classifier Trainer

Draw a rectangle ROI in a Color image, select either the Spot Meter, Color Map or Color Presence algorithm, and click on the Parameters button to open the Parameters window (Figure 66). This window lists the existing, compatible color classifiers. You can select a classifier from the list to apply to this ROI, or create a new classifier for this ROI. You can also select a classifier to edit or modify values. Creating or modifying a classifier opens the Classifier Trainer Tool.

A classifier created by this ROI is "owned" by this ROI. A classifier created by another ROI can be "shared" by this ROI.

The Color Map and Color Presence parameters also allows you to select and deselect individual colors defined by the classifier, for use in this ROI. A Color Classifier must already be defined for you to select colors.



Spot Meter - Statistical

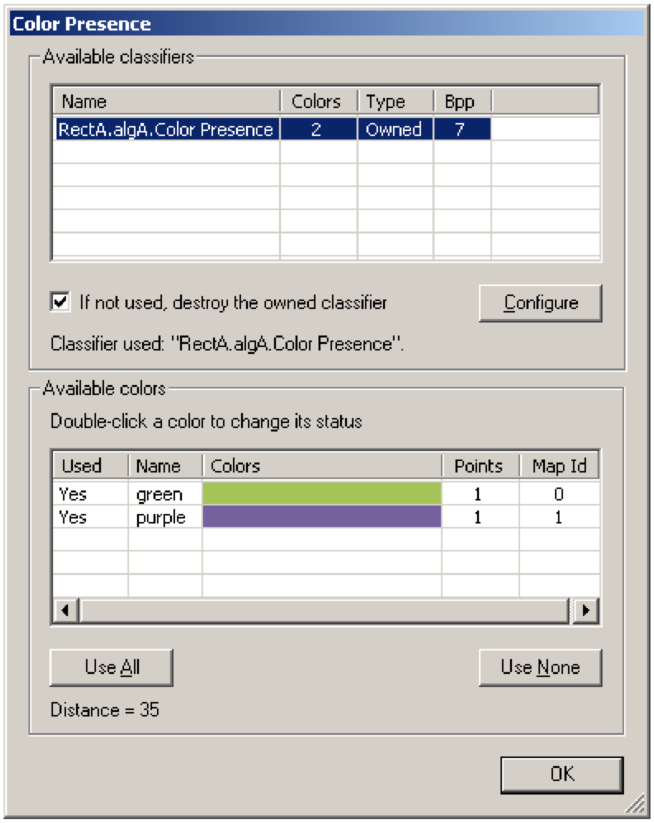
Available classifiers

Name	Colors	Type	Bpp
RectA.algA.Spot Meter	0	Owned	0

☑ If not used, destroy the owned classifier

Classifier used: "RectA.algA.Spot Meter". Configuration needed!

Figure 66. Spot Meter Parameters



Color Presence

Available classifiers

Name	Colors	Type	Bpp
RectA.algA.Color Presence	2	Owned	7

☑ If not used, destroy the owned classifier

Classifier used: "RectA.algA.Color Presence".

Available colors

Double-click a color to change its status

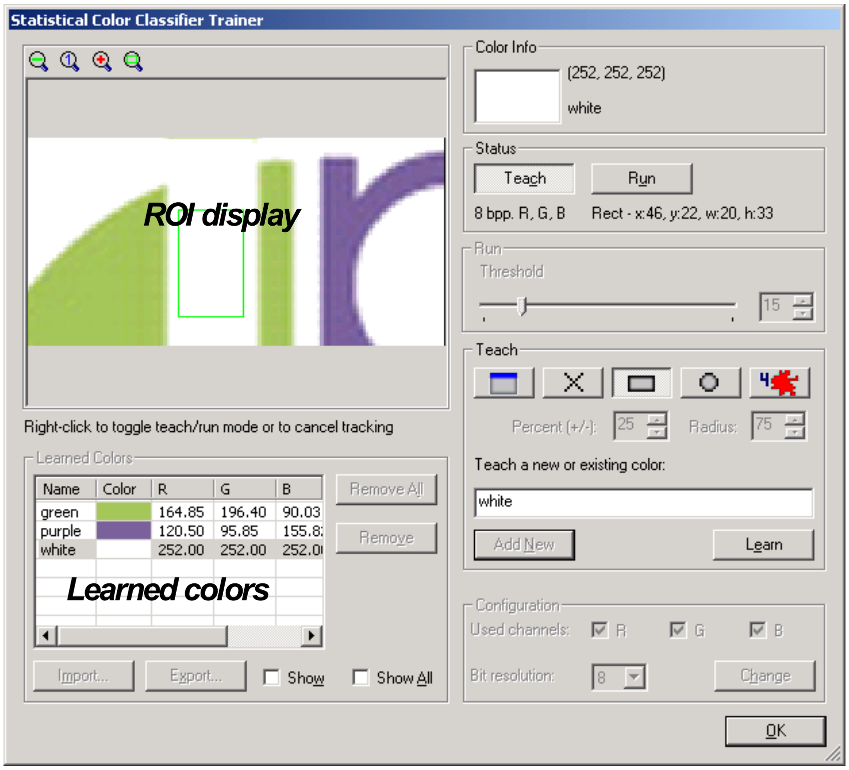
Used	Name	Colors	Points	Map Id
Yes	green	🟩	1	0
Yes	purple	🟪	1	1

Distance = 35

Figure 67. Color Presence Parameters

Click the "Configure" button to open the Color Classifier Trainer.

The Trainer window (Figures 68 and 69) creates or modifies the Color Classifier file. The file format is defined by the algorithm. The Spot meter creates a Statistical classifier. The Color Map and Color Presence algorithms create a Block classifier.



Color Info: (252, 252, 252) white

Status: 8 bpp. R, G, B Rect - x:46, y:22, w:20, h:33

Threshold: 15

Percent (+/-): 25 Radius: 75

Teach a new or existing color: white

Right-click to toggle teach/run mode or to cancel tracking

Name	Color	R	G	B
green	🟩	164.85	196.40	90.03
purple	🟪	120.50	95.85	155.8
white	⬜	252.00	252.00	252.0

☐ Show ☐ Show All

Used channels: ☑ R ☑ G ☑ B

Bit resolution: 8

Figure 68. Statistical Color Classifier Trainer

The ***ROI display*** can be zoomed in and out with the tools at the top of the display pane.

Learned colors - lists the data values for the trained colors in the current database.

- The Statistical Color Classifier (Figure 68) allows one color for each label or name, and allows a different Threshold value for each trained color.
- The Block Color Classifier (Figure 69) allows multiple color "points" for each label or name, and allows only one distance value for all colors.

Remove All - remove all colors in the database. Deleting a color takes a few minutes.

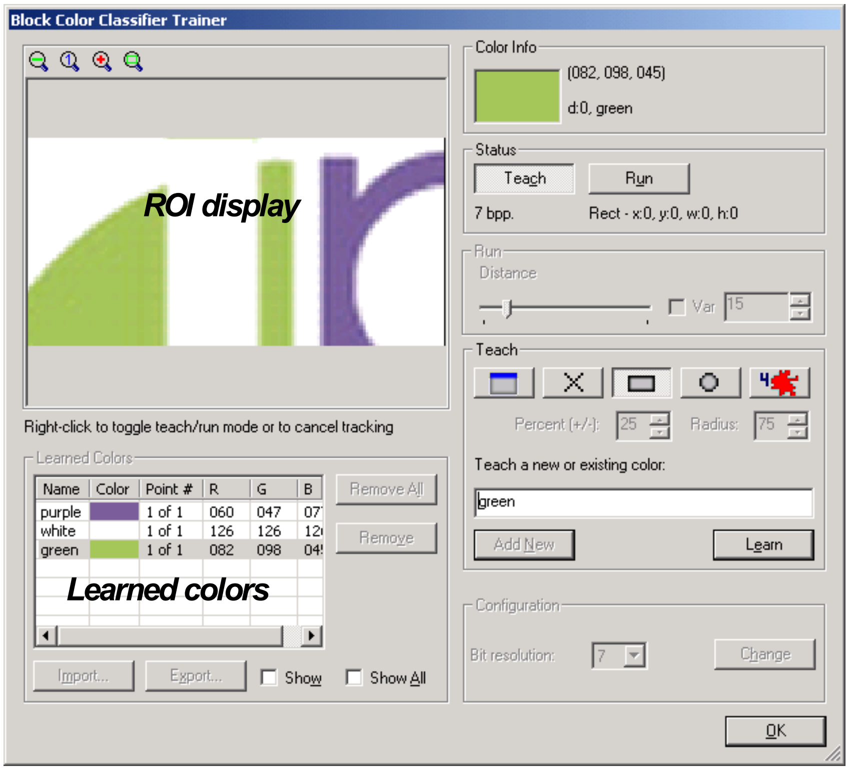
Remove - remove the highlighted or selected color from the database. Deleting a color takes a few minutes. The Block Classifier takes much longer to delete a color than the Statistical Classifier, because the whole color space model is recalculated.

Import - click to load a previously saved Color Classifier file. The files created by the Block and Statistical tools are not compatible.

Export - click to save trained Color Classifier file. The file extension is determined by the algorithm used to open the Color Classifier Training window. The files created by the Block and Statistical tools are not compatible.

Color Info - display information on the color at the cursor location in the display field. The 2 field or bipartite field shows the current RGB values in the top half. The bottom half identifies the trained color, if this color has already been trained.

Status - displays the teach or run mode, ROI size, and color depth. ***Teach/Run*** - click to toggle between teach and run modes. You can also right-click anywhere to toggle the mode.



Color Info: (082, 098, 045) d:0, green

Status: 7 bpp. Rect - x:0, y:0, w:0, h:0

Run: Distance □ Var 15

Teach: Percent (+/-): 25 Radius: 75

Teach a new or existing color: green

Right-click to toggle teach/run mode or to cancel tracking

Name	Color	Point #	R	G	B
purple	🟪	1 of 1	060	047	07
white		1 of 1	126	126	12
green	🟩	1 of 1	082	098	04

Configuration: Bit resolution: 7

Figure 69. Block Color Classifier Trainer

Distance (Block only) - controls the color's distribution around the trained color point. This value is applied to all colors in the Block Color Classifier.

Threshold (Statistical only) - controls the color's distribution around the trained color point. Each color has its own threshold value in the Statistical Color Classifier.

Teach - the buttons become active when you are in Teach mode.

- *ROI:* use the entire ROI in the display pane.
- *Point:* create a point inside the ROI for training a color.
- *Rectangle:* draw a rectangle inside the ROI area for training a color.
- *Disc:* draw a Circle Area inside the ROI for training a color.
- *Growth:* grows an area (flood fill) of included colors starting with the point where you click in the ROI, limited by the percent (a color value tolerance) and radius (a maximum size) and the Threshold or Distance. Neighbors are horizontal and vertical only (4-way sampling), not on the diagonal.

Teach a new or existing color - type a name for the color.

Add new - adds the name to the database and opens entry fields for teaching.

Teach - samples the color in the ROI or sub-ROI, and fills in the database entry.

Configuration - set the channels and color depth before training colors. Changing the values in this area will delete all entries in the current database.

Bit resolution - bits per color channel used in the database.

Channel (Statistical only) - select or deselect color channels used in the database.

5. Camera Calibration

"Camera calibration" refers to establishing a coordinate transformation between the pixel coordinates of the camera and some real world coordinate system. This permits accurate measurements to be made in real units from any image window. Sherlock has an extremely flexible, easy-to-use, facility for creating and using such transformations.

How Camera Calibration Works

Sherlock presents three main methods of calibration: Points, Scale, and Grid. Point calibration, or "Calibration using Points" uses four points in a single image. If the four points in the camera image are associated with real world coordinates or distances, Sherlock can use internal mathematical conversions to establish coordinate transformations that correct for "affine transformation" or small changes in scale, offset, and rotation (or skew), plus "linear perspective distortions" when the plane of the camera is not parallel to the plane of the object; straight lines in the object are still straight in the image, but parallel lines are no longer parallel. See Figure 73 (page 95) for more information on perspective distortion.

Scale calibration, transform, or "Direct Calibration" uses one horizontal and one vertical multiplying (scale) factors. This method only changes the pixel or camera coordinates, and does not compensate for perspective distortion. This method performs only limited affine transformation.

Grid calibration uses a checkerboard, or two-dimensional grid, to establish a four point mapping, or multiple localized corrections to the coordinate system. This method works with the assumption that the checkerboard or grid should consist of (only) straight perpendicular and parallel lines that define square cells. Localized 4 point coordinate transformations are established. The radial perspective method assumes the cells are uniform in size and performs non-linear distance corrections from a central point.

Once a calibration has been established, it may be used in any number of other image windows, by using the *Active Calibration* option under the Image Window Options. This allows a single calibration to be used by many image windows. Sherlock also supports defining more than one calibration, and using a different calibration system in different image windows; for example, when an investigation uses more than one camera. Each camera view should be separately calibrated.

When a calibration image window is recalibrated, all image windows referencing that calibration will automatically be updated. Sherlock performs calibration on all subroutines enabled for calibration.

All Point and Line readings are represented in the coordinate system of the image window for which they were defined. Thus, once an image window is calibrated, dimensions will read out in real units, angles will be relative to the real world X axis, and even circles will be computed as ovals if required due to perspective distortion.

You can convert from Camera Coordinates to Calibrated coordinates with the instruction "Pixels To World". You can convert from Calibrated coordinates to Camera Coordinates with the instruction "World To Pixels".

Setting Up a Camera Calibration

There are two ways to set up a calibration, programmatically, usually in a Subroutine, and interactively (manually) from the Image Window Options. Each of these also supports using points, or using scale multipliers.

Programmatic Calibrations:

- Calibrate Using Points - uses four points at known distance or separation.
- Calibrate Using Point Arrays - uses an array of points with known locations or separation distances.
- Calibrate Using Transform - uses scale multipliers.

Interactive Calibrations:

- Direct Calibration - uses scale multipliers.
- Point Calibration - uses four points at known separation distance.
- Grid Calibration - uses a checkerboard with squares of known size, or a grid of lines or spots (or holes) at known separation distance.

Program Calibration Using Points

Programmatic Calibration is usually performed in a subroutine, because you only need to run calibration or recalibrate occasionally or infrequently. The following steps give a typical example of how camera calibration is established and used. This requires using a part or picture that has 4 identifiable points that are separated by a known distance or measurement. You could use a grid of know size, or a part that you know the dimensions of. An ideal target is four circles or holes spaced out along the corners of a rectangle. Try to fill as much of the field of view as possible to maximize accuracy.

Area ROIs with thresholds and Centroids work reliably in delivering true sub-pixel accuracy. Your "points" should be relatively large (40 pixels diameter is typical) to minimize errors due to "fringe" pixels.

For holes or circles, spoke ROIs and the algorithm First Edge combined with the instruction "Points to best Circle" works reliably. Your circles or holes should be relatively large (40 pixels diameter is typical) to minimize errors due to "fringe" pixels.

1. Create a new investigation file. In the Program window, create a subroutine, by clicking on the "SUB" button in the Program window toolbar. Click on the name "SubA", Press F2 or click on "SubA" a second time, to open the text field. Type **Calibrate** and press return.
2. Double-left-click on the subroutine name, to open the Subroutine Edit dialog. Select "Execute on calibration". Click OK to close the dialog.
3. Create an image window in this Subroutine, by clicking on the "ACQ" button in the Program window toolbar.
4. Open the Image Window Options for the new image window (imgB).

This example uses an image file. You would use a part or picture in the camera's field of view, that has four known coordinates (in the same plane). An ideal target is four points spaced out along the corners of a rectangle.

5. Click on the "Image Source" tab, if it is not already selected. Click the radio button beside "File" if it is not already selected. Click on the browse button and find the file "Calibration_Example.bmp" found in the "Example Images" directory on the Sherlock CD or on the Vision Appliance.
6. Click on the "Calibration" tab. Click on the "Add" button. "CalibrationA" appears in the list of "Available system calibrations". (You can add a Calibration system for each camera.)
7. In the image window, create four ROIs to find the four points. For this example, use four Spoke ROIs to find the edges of the four circles. Click on the "Spoke" button in the Image Window toolbar. Click twice in the center of one of the holes, then click outside the hole. Repeat for the four holes indicated in Figure 69.
8. Test the repeatability of finding the four points, by pressing the F11 key, or by pulling down the "Run" menu and selecting "Calibrate".
9. Double-left-click on the first ROI in the Image window, to open the ROI Properties. Select the "First Edge" algorithm. Repeat this step for the other 3 ROIs.

10. Right-click on the first ROI in the Program window. Move your cursor down the pop-up menu, to "Geometric". In the submenu, select "PtsToBestCircle". Drag the output Point Array "P[] edge point" from the first Spoke ROI, and drop it on the input point array "P[] pts((...) [0])" under the PtsToBestCircle Instruction. Repeat this step for the other 3 ROIs and 3 instructions.

11. Hover the cursor over the edge point array for the first ROI "P[] edge point". Several point values should be displayed in a temporary pop-up window. These are the edges found along the spoke lines of the ROI.

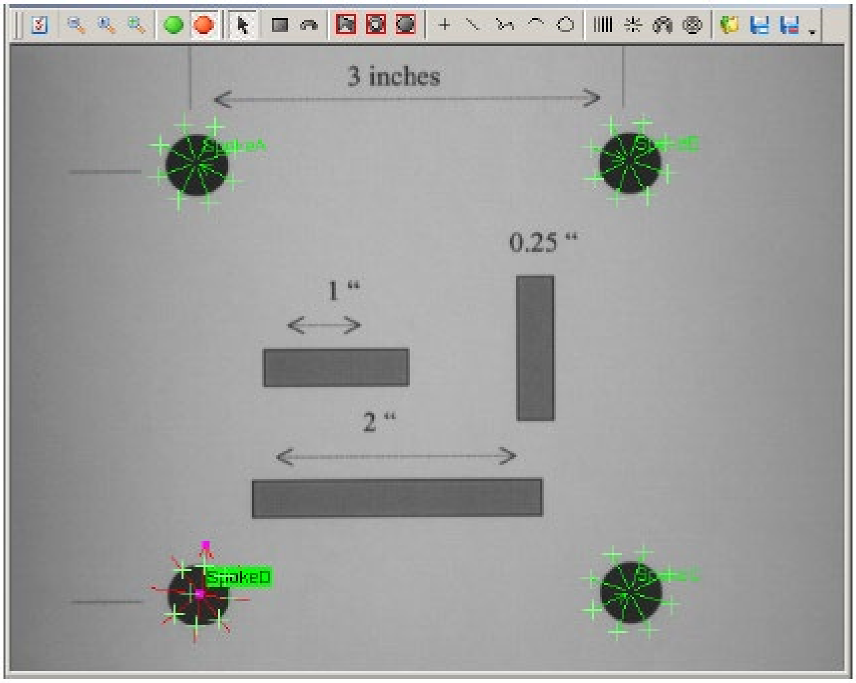


Figure 70. Four Spoke ROIs in calibration_reference.bmp

12. Right-click on the last instruction in the subroutine. Move your cursor down to "IO:Calibration" and select "Calibrate Using Points" in the submenu.

13. Double-left-click on "HDL calibration <EMPTY>". In the "Select Input" dialog, use the drop list to select "CalibrationA" and click OK to close the dialog.

14. Drag the center point "P center" from the first "PtsToBestCircle" instruction, and drop it onto the first point input "P pixel0" under the calibration instruction. Repeat for the other 3 center points and point inputs in this instruction. The order is not important, but you do need to know which point is which.

15. Double-left-click on "P world0". The default value is 0.00,0.00. Assume the first point is going to be the 0,0 origin of your calibrated image. Click OK to close the dialog.

16. Double-left-click on "P world1". Enter the "real measurement" or "real world" coordinates for the second point. If for example, this point is horizontally 3 inches to the right of the first point, enter 3,0.

17. Double-left-click on "P world2". Enter the coordinates for the third point. If for example, this point is vertically 3 inches below the first point, and 3 inches to the right, enter 3,3.

18. Double-left-click on "P world3". Enter the coordinates for the fourth point. If for example, this point is 3 inches below the first point, enter 0,3.

19. Press the F11 key, to run the calibration routine.

20. At the bottom of the Program window, click on the Main tab. Double-left-click on the image window instruction "imgA" to open the Image Window Options dialog. Click on the Calibration tab.

21. Use the drop list beside "Active calibration" to select "CalibrationA". Click in the box beside "Display grid" and change the "grid size" to 1. Click OK to close the dialog. The image window displays a grid, with a white circle to indicate the origin or 0,0 point.

NOTE *If the points on your part or picture defined a rectangle, and if the camera was perpendicular to the picture or part, you should see a regular, rectangular grid. If not, recheck the values for the "world" points. It is very easy to switch X and Y when entering values. Also check the angle of the camera relative to the picture or part. If you used the Calibration_Example.bmp file, and entered the numbers given, the grid will not be rectangular.*

To remove the calibration from an image window, open the Image Window Options, click on the Calibration tab. Use the drop list beside "Active calibration" to select "[None]".

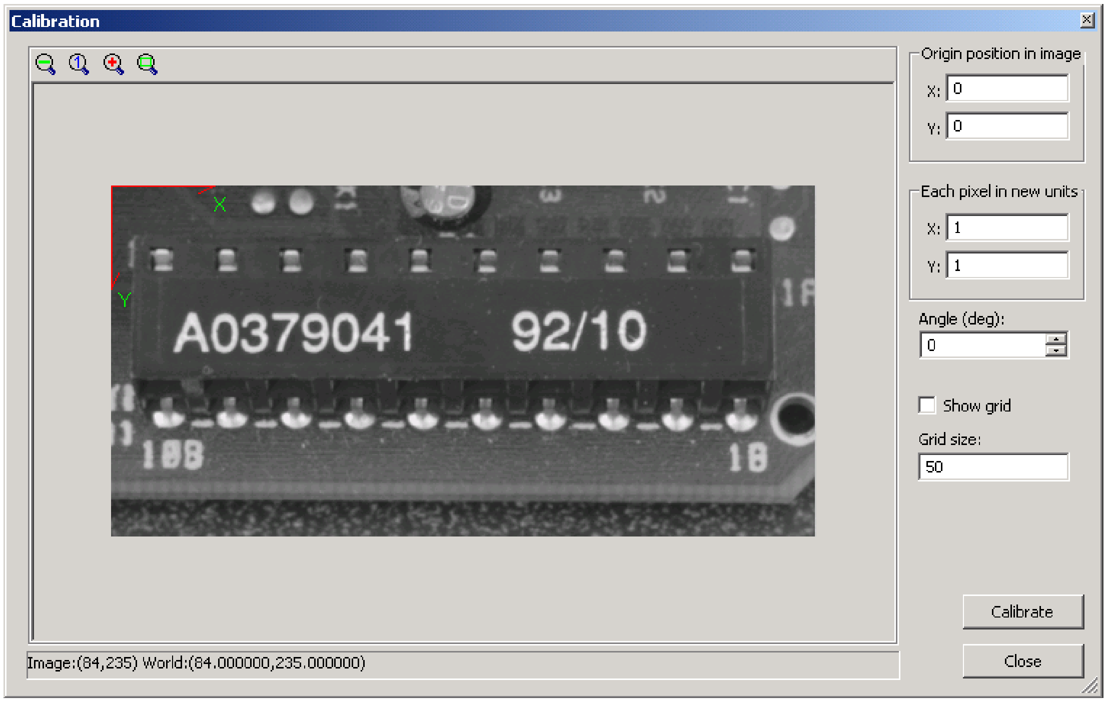
Interactive Calibration Using Transform

Direct calibration uses scale multipliers. This requires an image with a part that you know the size of.

1. Create a new investigation file. Pull down the Options menu and select Application.
2. On the second line, beside "Displayed floating point precision" enter 6. Click Close.
3. Open the image Mark1.bmp. For this example, we assume the IC Chip in the picture is 10 mm wide by 3 mm high.
4. Move the cursor to the left edge of the IC Chip. The status bar shows the cursor position in pixel coordinates; approximately x:00014 y:00074.
5. Move to the right edge of the IC Chip. The status bar shows the new cursor position in pixel coordinates; approximately x:00455 y:00066.
6. Move to the top edge of the IC Chip. The coordinates are approximately x:00271 y:00035.
7. Move to the bottom edge of the IC Chip. The coordinates are approximately x:0263 y:00133.

The width of the Chip is 455 - 14 = 441 pixels. The height of the Chip is 133 - 35 = 98 pixels.
And 10 mm / 441 pixels = 0.022175. 3 mm / 98 pixels = 0.030612.

8. Double-left-click in the image window, to open the Options dialog. Click on the Calibration tab. Click the "Add" button.
9. Click on "CalibrationA" in the list of "Available system calibrations". Click on the "Direct Calibration" button.



Calibration

Origin position in image

X: 0

Y: 0

Each pixel in new units

X: 1

Y: 1

Angle (deg): 0

□ Show grid

Grid size: 50

Image:(84,235) World:(84.000000,235.000000)

Figure 71. Direct Calibration

10. Under "Each pixel in new units" enter 0.022175 beside "X" and enter 0.030162 beside "Y".
11. Click the "Calibrate" button. Click Close.
12. In the Image Window Options dialog, use the drop list beside "Active calibration" to select "CalibrationA". Click OK.
13. In the image window, click the "Line" button in the image window toolbar. Draw a line below the characters printed on the chip, starting from slightly left of the chip and ending slightly right of the chip.
14. Double-left-click on the Line ROI, to open the ROI Properties dialog.
15. Select the "Threshold" preprocessor. Change the threshold to 90. Click OK.
16. Select the "Inside Caliper" algorithm. Click Close.
17. Run the investigation once.
18. In the program window, hover the cursor over "dist between edges". The value should be between 9.7 and 10. This is the calibrated distance (approximately 10 mm).

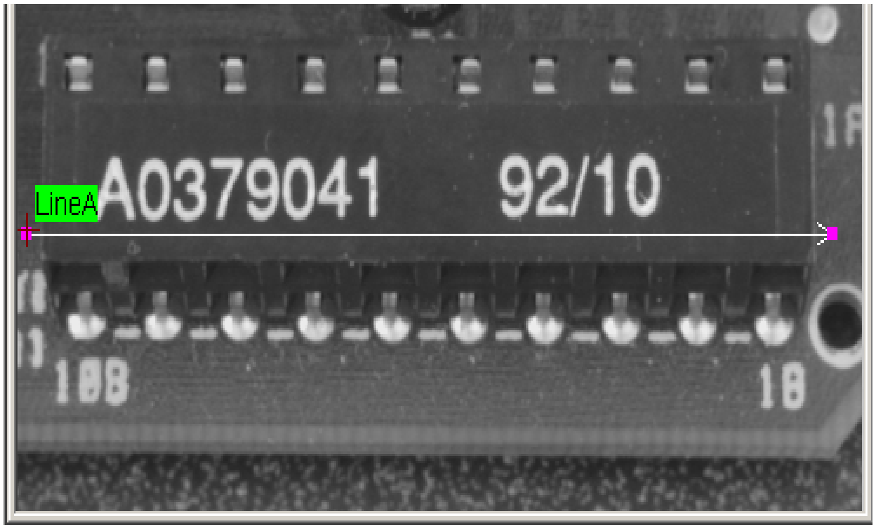
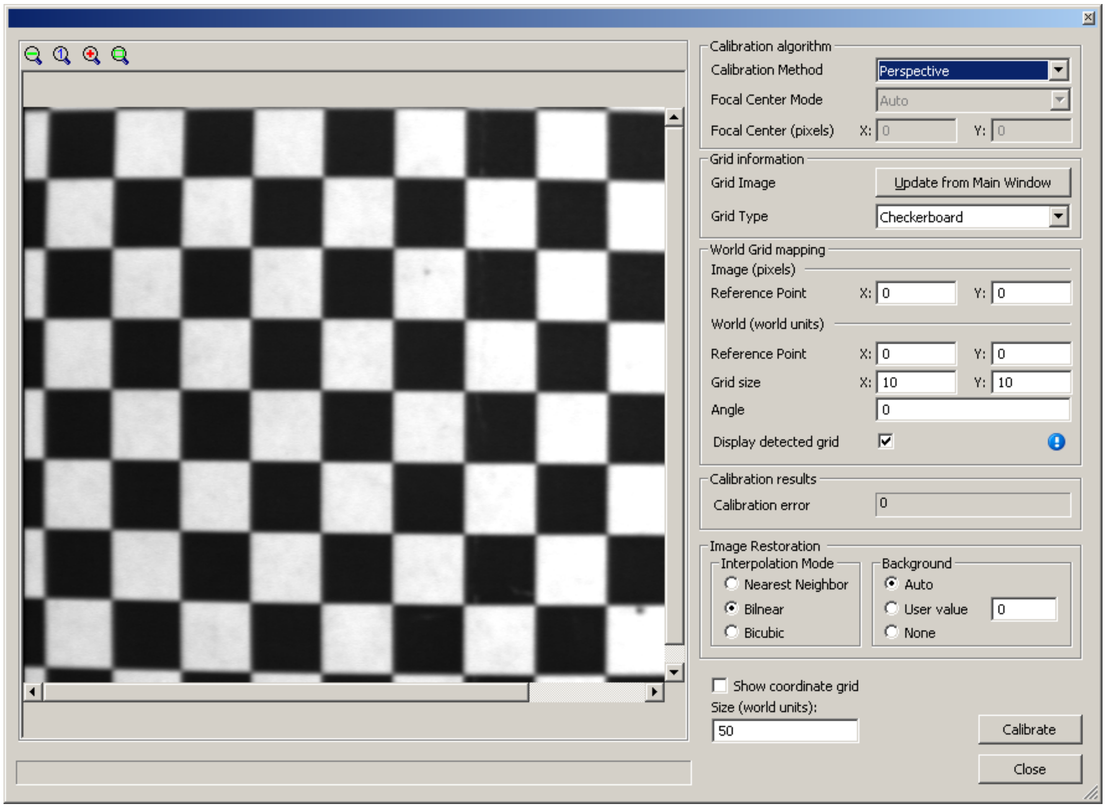


Figure 72. Line ROI

Interactive Calibration Using Grid

Grid calibration uses a grid or checkerboard. A checkerboard pattern is included at the end of this manual. For high accuracy measurements, you can purchase precision reference grids.

1. Place the 10 mm checkerboard in front of the camera, at the same distance as the part you are measuring.
2. Create a new investigation file, or open a file that you have already created.
3. If the checkerboard does not appear in the Image Window, click on the green "live" button to acquire an image. For best results, the checkerboard should fill the image area, if possible.
4. Open the Image Window Options (there are 3 ways to open this menu: use the icon on the Image Window toolbar, or the Image Window menu in the main window, or double-click in the Image Window).
5. Click on the Calibration tab.
6. If there is no entry in the "Available system calibrations" list, click the "Add" button to create a new calibration system. "CalibrationA" should appear in the "Calibrations" list under "Available system calibrations".
7. Highlight (click on) "CalibrationA" and then click the "Grid Calibration" button. This will open the Calibration window.



Calibration algorithm

Calibration Method: Perspective

Focal Center Mode: Auto

Focal Center (pixels) X: 0 Y: 0

Grid information

Grid Image: Update from Main Window

Grid Type: Checkerboard

World Grid mapping

Image (pixels)

Reference Point X: 0 Y: 0

World (world units)

Reference Point X: 0 Y: 0

Grid size X: 10 Y: 10

Angle: 0

Display detected grid ☑

Calibration results

Calibration error: 0

Image Restoration

Interpolation Mode: ○ Nearest Neighbor ● Bilnear ○ Bicubic

Background: ● Auto ○ User value 0 ○ None

□ Show coordinate grid

Size (world units): 50

Figure 73. Grid Calibration

8. In the drop list beside "Calibration Method" select Perspective. This method also corrects for linear perspective distortion across the image, and is the fastest of the available methods.
9. In the drop list beside "Grid Type" select Checkerboard.
10. Under "World (world units) enter 10 in the fields beside "X:" and "Y:" (the squares are 10 mm.)
11. The box beside "Display detected grid" should be checked.
12. Click on the "Calibrate" button. Wait for the display to show a pattern of green lines outlining the squares of the checkerboard (this is the "detected grid").

The image is now calibrated, and distance measurements will be reported in millimeters.

13. If you need to define a point of origin, to give positive and negative distances, click on a point in the image where you want the origin to be. Under "Image(pixels)" the X and Y number fields beside "Reference Point" should show the X and Y coordinates in pixels (not calibration units).
14. Click the "Calibrate" button again, to assign the new origin (0,0) point to the calibrated image.
15. Click the "Close button"
16. In the Image Window Options menu, (calibration tab) use the drop list beside "Active calibration" to select "CalibrationA".
17. Click "OK" at the bottom of the Options menu to close the Image Window Options.
18. Remove the checkerboard pattern from the camera view.

NOTE *Calibration and Restoration are separate operations, and cannot be done at the same time. Perform a calibration in one Image Window. Select the completed calibration as Active, for performing restoration in a different Image Window. The two windows can use the same image source.*

Grid Calibration Methods

The grid calibration utility includes four algorithms, each optimized for a particular type of distortion.

Perspective corrects for perspective distortion only; it does not correct for localized distortion.

Piecewise Bilinear corrects for localized distortion only; it does not correct for perspective distortion.

Piecewise Perspective corrects for localized and perspective distortion (Perspective and Piecewise Bilinear combined).

Radial corrects for radial distortion only; it does not correct for other forms of localized distortion or for perspective distortion. Radial distortion, which is introduced to some extent by most lenses, is the same along any line drawn from the center of the image (the center of the lens) to the edge. Pincushion and barrel ("fisheye") distortion are examples of radial distortion.

The different calibration methods - Perspective, Piecewise Bilinear, Piecewise Perspective, and Radial - take different amounts of time to create correction maps. But once their correction maps have been created, they all take the same amount of time to correct an image.

The different image restoration methods - Nearest Neighbor, Bilinear, and Bicubic - take increasing amounts of time to remap the pixels, but are also increasingly accurate.

Multiple Calibrations Per Camera

Sherlock calibrations are valid within one plane of measurement. If a single camera is to be used to make measurements in multiple planes, multiple calibrations are required. Create one calibration image for each measurement plane required, and teach target points for that plane. When making measurements in other images, simply select the appropriate calibration system for the ROIs in that plane of measurement required.

Automatic Recalibration

Sherlock provides a powerful calibration facility that supports automatic recalibration in the field. The calibration subroutine stakeouts form an investigation of their own that may be used to locate predetermined points on a custom calibration target, or a piece part that has known dimensions. At any time, an operator or manufacturing engineer can insert this test part, press F11, and recalibrate all of the cameras.

NOTE *Camera calibration is saved with the .IVS file, so remember to save after recalibrating.*

Perspective Distortion

When the camera sensor plane is at an angle with respect to the object plane, “perspective distortion” or “skew” occurs. The squares of a checkerboard would look like trapezoids, as shown in Figure 74. Perspective distortion does not cause straight lines in the object to bend or curve in the image.

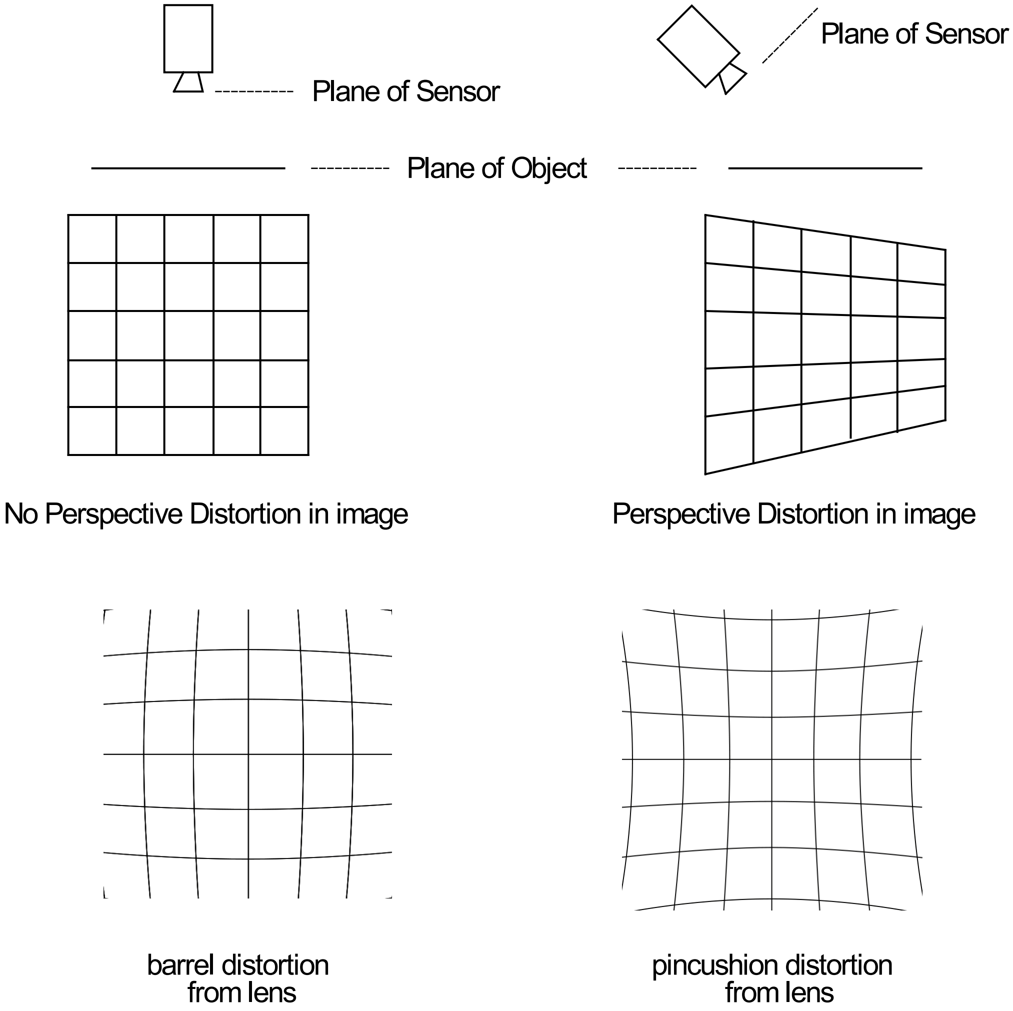

Plane of Sensor
Plane of Sensor
Plane of Object
No Perspective Distortion in image
Perspective Distortion in image
barrel distortion from lens
pincushion distortion from lens

Figure 74. Distortion Effects

6. Alignment

Frequently in machine vision, parts are not repeat-ably positioned with respect to the camera. In many cases, this misalignment may be compensated for by using relative measurements; for example, Sherlock may find the length of a part by using line ROIs to find each end, and then a Geometric instruction to measure the distance between the two end points. In this way, the measurements will remain constant even if the part shifts around a bit, as long as it does not move so far that one end falls off the end of the corresponding line ROI.

In some cases, however, absolute registration is required. If a "Golden template" is to be used, or if a small feature must be measured, it may be impossible to oversize the Areas and Lines enough to guarantee robust performance regardless normal variation in part position. In these cases, Part Alignment can be used to easily solve the registration problem.

How Alignment Works

The Sherlock Part Alignment feature relies on *Landmarks* to pull ROIs back onto the part locations at which they were originally taught. All point readings may be used as Landmarks. Landmarks mark point locations returned by Instructions, drawing the memorized point in a target or cross-hair.

At the time a point reading is marked as a Landmark, it's current location is memorized. This point is then redrawn with a large cross-hair indicating the position of the memorized point.

Once the Landmarks have been located, all ROIs enabled for Alignment are moved according to the number of Landmarks found:

1 Landmark: ROIs are translated in X and Y

2 Landmarks: ROIs are translated in X and Y, and Rotated.

3 Landmarks: ROIs are translated in X and Y, and Rotated.

After the ROIs have been repositioned, inspection of the image window continues from the point just after the definition of the last Landmark.

ROIs are always drawn in both their original and repositioned locations. A dotted line is used to show original locations, and a solid line is used to show the repositioned location.

Once an alignment has been found in one image window, it can be reused in other image windows that reference the same part using the ***Alignment*** option under ROI Options.

In Sherlock 6, the whole image window (Stakeout) was enabled for landmarking. In Sherlock 7, each ROI is separately enabled for alignment, and moves within the image window.

One Point Landmarking

In one point landmarking (a single landmark is defined in a image window), the offset from the original landmark position to the one found at run time is used to translate all ROIs following the landmark in X and Y. No rotation is possible with a single landmark.

Two Point Landmarking

In two point landmarking (where exactly two landmarks are defined in a image window), the offset from the original landmark positions to the ones found at run time are used to translate and rotate all ROIs following the second landmark. The points must be fixed points with respect to the part under test; for example, corners, centers of alignment holes, etc. The distance between the two points should always be constant, regardless of point orientation.

Three Point Landmarking

Three-point landmarking assumes that the first point is a fixed point relative to the part, and the two following points are only used to compute rotation information. The original angle between the points is remembered, and at run time the new angle is found, compared to the original, and used to calculate a rotation distance.

Setting Up Part Alignment

The following steps give a simple example of how Part Alignment is used. For the example, we will assume that translation AND rotation are possible, and so we will use two Landmarks.

1. Create a image window. Place a sample part in the field of view and select two orientation points that may be reliably located. Let's assume that two features of the part will be found using two ROIs running the Centroid algorithm.
2. Teach the two Landmarks by creating small Rectangle ROIs over the two features on the part. Open the ROI Options dialog, select the Centroid algorithm. Repeat for the second ROI. Run the investigation a few times to verify that the Centroid algorithm reliably locates the features.
3. Open the Image Window Option dialog, and click on the "Alignment" tab. Click the "Create Alignment" button.
4. Drag the two points onto the "alignmentA".
5. Create additional ROIs as desired to align with various features of interest on the part. These ROIs will be used for actual inspection, not just alignment.
6. Open the ROI Properties Dialog. Click on the "Properties" tab. Use the drop list to select "alignmentA".
7. Move the part around and run the investigation. If the centroid ROIs successfully locates the part features, the other ROIs will be moved into the correct position relative to the part body before inspection.

Landmarking with Search

The three Search algorithms return a point and an angle for the location of the pattern. The Search algorithms also return two more points for the first found occurrence of the pattern, called "pattern left" and "pattern right". These two points are located to the left and right of "best point" at a distance of half the width of the pattern.

Use a Search algorithm to find your pattern. Open the Image Window Options, click on the "Alignment" tab. Drag the three points: "pattern left", "best point" and "pattern right" onto the alignment.

Earlier versions of Sherlock 7 (before 7109-SP2) did not return "pattern left" and "pattern right". It was necessary to use the Geometric instruction MakeAnchor, to create a pair of points from the "best point" and "best angle" returned by the Search-Geometric.

Retraining Alignment

To re-teach or retrain Landmarks

1. Move a Landmark point to the new location
2. Open the Image window options,
3. Click on the Alignment tab,
4. Click on the Alignment system you wish to retrain,
5. Click on the "Reset home position" button.

Disabling Alignment

After Part Alignment is in use, it can be disabled by:

- Each ROI - Opening the ROI Properties, clicking the Alignment tab, and selecting "None" in the drop list under alignment.
- Entire image or investigation - Opening the Image Window Properties, clicking the Alignment tab, and delete all points under the alignment, or delete the Alignment.

7. Search Algorithms

Pattern Finding Tools

A Pattern Finding tool, is a machine vision tool that measures and reports the location of a previously trained pattern in an image. A pattern is a region of interest (in an image) that contains meaningful and unique information, as in Figure 75. A pattern can be as simple as an edge with few pixels, or a very complex pattern of thousands of pixels, such as human face. Typically, a pattern is trained once and searched for in other images. The pattern can be saved with its search parameters so it can be loaded and used at any time without the need to retrain.

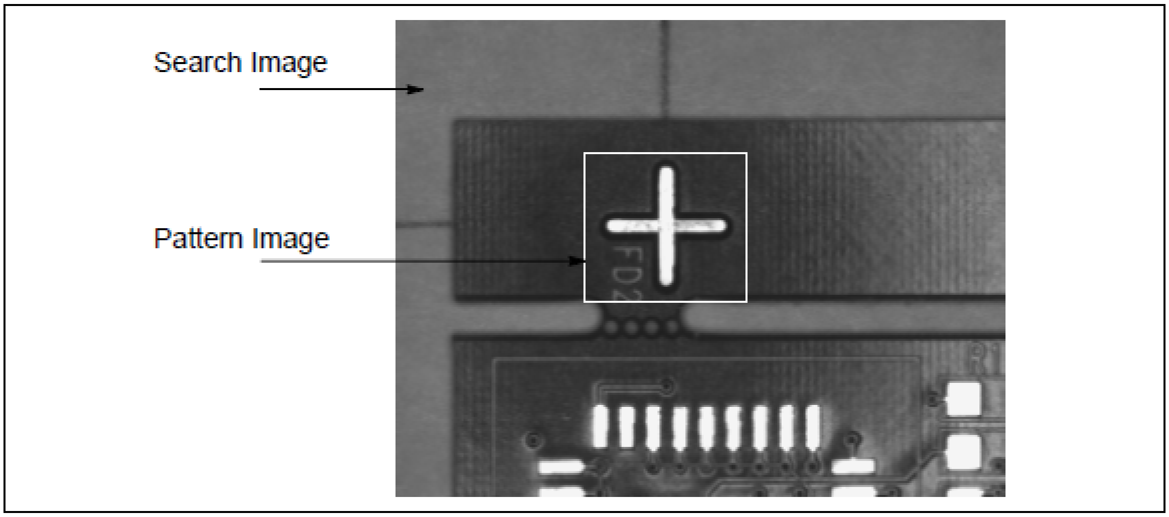


Figure 75. Region of Interest as a Pattern Image

Correlation Search

The Correlation Search is the "traditional" texture-based search. The pattern is defined by its gray-scale intensities. The search process compares the intensities in the Trained Pattern to the possible instances in the Search Image. The search strategy is described as a "spiral". The possible locations of the pattern are located using very few pixels of the trained pattern or model. Each of the possible locations are checked again with more pixels (subsample) of the pattern and the locations with low score are removed. The process is repeated with a larger number of pixels from the trained pattern, until an instance of the pattern is located, to a single pixel accuracy. After that, one of the sub-pixel search kernels takes over, to locate the position of the pattern in sub-pixel accuracy. 1/64 of a pixel is common for correlation-based search if the pattern is not degraded.

Figure 76 shows a Trained Pattern and possible matches. Each of the possible matches are reexamined with finer steps and repeated sampling, until the best match is found.

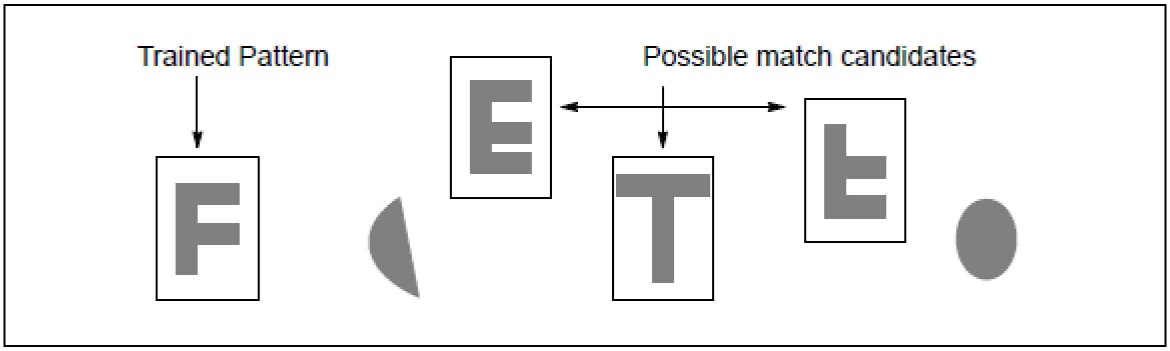


Figure 76. Match Candidates

SubPixel Kernel

The Subpixel Kernel performs neighborhood computation to calculate the pattern position in sub-pixel accuracy. This kernel has a capability of 1/46 pixel accuracy, given a good pattern. The accuracy drops proportionally with pattern degradation.

Geometric Search

Geometric Search is a pattern finding tool that is capable of locating objects in any orientation (0 to 360) without sensitivity to variations in lighting, part reflectance, background noise, occlusion, and missing portion. The speed of locating an object is independent of the orientation of the object. Objects can be occluded, overlapped or partially outside the ROI.

Geometric Search is a contour-based algorithm. This means that it cannot perform well when the trained or search object contains **only** a small segment of contours that is less than 120 pixels long.

Concept

Geometric search is a contour-based algorithm that can locate objects based on their geometrical contours. There is no use of texture in the algorithm.

During training time, the algorithm extracts useful contours from the image, and uses their curvature information as the reference of comparison. The algorithm allows you to specify minimum and maximum contour lengths, to reject information in the image.

During search, the algorithm extracts most of the contours in the image, then compares them with the reference contours. The results of the comparison are the location, angle, scale, and score of best-matched contours. Contours are compared by their curvatures only.

Line Based Search

Line based Search is another form of Geometric search, a contour or edge based search algorithm. Where Geometric search finds all the edges of the object, Line base search finds the edges and discards all contours that are not straight lines. Only the straight lines in the object are used for pattern matching. The algorithm allows you to specify minimum and maximum contour lengths, to reject information in the image.

There are two great benefits to imposing this limit on the object and pattern. The algorithm is faster than the geometric search, and is not sensitive to scale changes. The line based pattern can be larger or smaller (nearer or farther) in the search area.

Training A Pattern

Regardless of the search method used, the model image or trained pattern has to be unique in the search image. The pattern has to have defined edges for Geometric Search to perform well. The pattern has to have defined gray-scale values (fill) for the Correlation Search to perform well. For best results, train on a good pattern and a worst case pattern, and compare the setup parameter values for both cases.

For some very special types of non-linear reflectance images, you can train a pattern using both Geometric Search and Correlation Search and use both techniques to search for the pattern. This increases the level of robustness, but also increases search time.

Figure 77 shows unique, less unique and not unique patterns. If using the "less unique" pattern with Geometric Search, increasing the Edge Strength to include the pinhole trace would improve the results.

1. Draw a Rectangle ROI in the Image window, to enclose the unique pattern or feature.
2. Open the ROI Properties. Select a Search algorithm.
   The algorithm immediately trains on the pattern in the ROI, when you select a search algorithm.

3. Open the Parameters and make adjustments as needed. Close the parameters. Retrain if needed. After retraining change the run mode to Run.
4. Increase the ROI to enclose the area where the pattern may appear, or the entire image area.

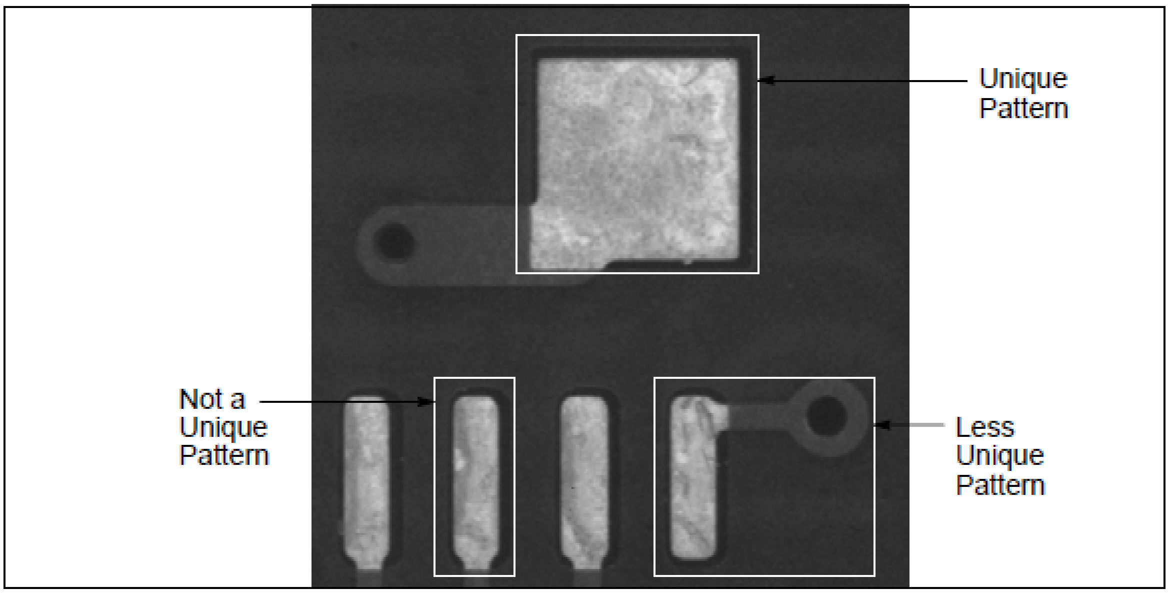


Figure 77. Pattern Uniqueness

Choosing Between Correlation and Geometric Search or Line Search

In graphics and drawing there are properties called outline and filling (texture). For example, a drawn rectangle has an outline and a filling color. If the filling color is the same as the background color, the rectangle "has no filling". The filling color has no boundary or contour of its own. See Figure 78.

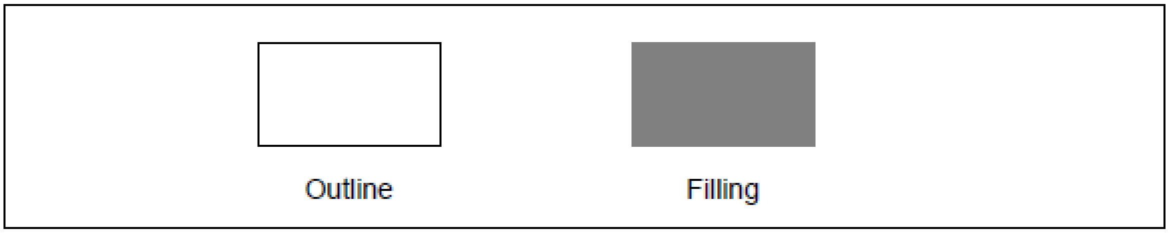


Figure 78. Outline and Filling

In real-world images, the outlines are the edge pixels, or a sudden transition of the gray-scale value in neighboring pixels. For the image of a fiducial in Figure 79 ("original image") both search techniques could be used. The image has enough filling to allow significant sub-sampling (using less pixels) which makes the Correlation Search technique a viable solution. The pattern also has a very clear outline, which makes the Geometric Search technique a viable solution.

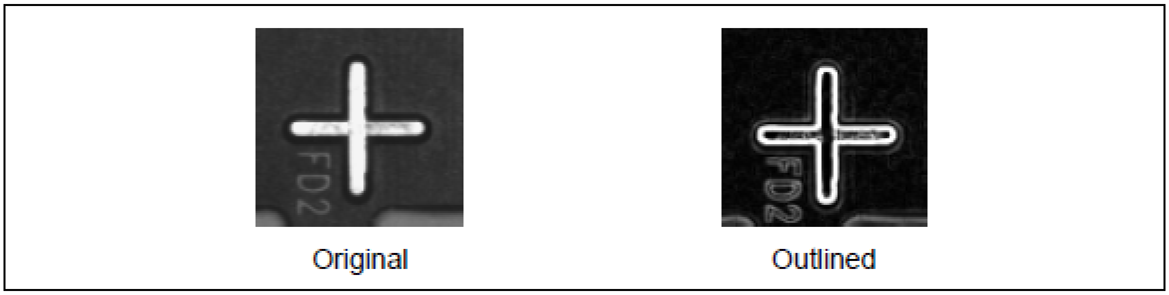


Figure 79. Fiducial and Outline

As shown in Figure 80, with a zoom factor of 8 the pattern is still recognizable. This makes Correlation attractive to use for its speed and robustness. A zoom by 8 provides speed up to 4x4x4x4 = 256 times faster than using every pixel in the complete pattern.

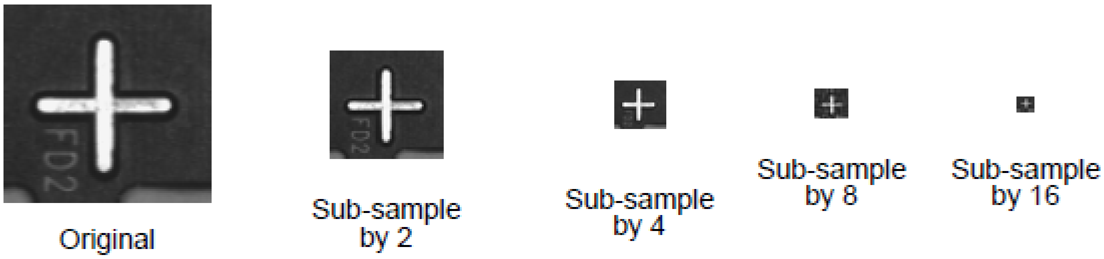


Figure 80. Fiducial and Zoom Factors

Figure 81 shows an image that does not have good prospects for Correlation Search. With a zoom factor of 2, the pattern starts to lose its content. With a zoom factor of 4, the pattern has disappeared.

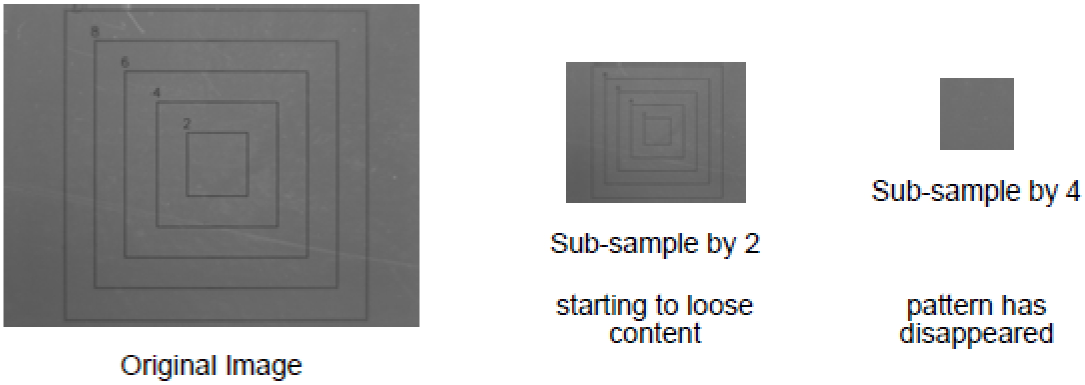


Figure 81. Contour and Zoom Factor

The image in 180 has part of the fiducial missing. Both Geometric Search and Correlation Search could easily find the fiducial in Figure 82, however, the position accuracy for the Geometric Search is very high (exceeds 1/60 of a pixel), while the position accuracy of the Correlation Search would drop considerably based on the size of the missing part. 2/10 of a pixel to 1 pixel is not an uncommon accuracy drop for the Correlation search. Other kinds of defects, such as occlusion, debris, and oxidization have the same effect on the results.

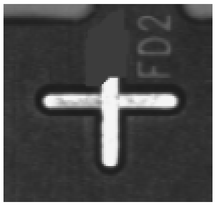


Figure 82. Fiducial with Missing

For a contour image (like Figure 81) the pattern or shape is defined by a contour, with edges that are typically only a couple pixels in width or thickness. The Correlation search would have difficulty finding that pattern, and if it finds the pattern, it will take considerable time. The problem becomes very serious when there is no pattern. The Correlation search would return that it found the pattern, but in reality there could be nothing but the background. Since Geometric Search uses the relationships between contour pixels, finding the contour pattern is a very easy task. In contour type of images, the Geometric Search should be used. In a blurred image, or in images where edges are not defined, the Correlation Search is the preferred choice.

Image Defect and Variation

Changes occur in a part, caused by use (wear) or by manufacturing process. Some common examples are changes in reflectance, integrity (missing part of the pattern), scale, and mechanical positioning. The light source used to illuminate the part could change with age, or the intensity of the light source may vary with respect to the position of the part. All of these would make the search pattern differ from the trained model. You want your Search algorithm to find the part by ignoring the particular defects or variations. The following illustrations show several different kinds of image variation and defects.

Figure 83 shows trained model, and possible variations due to illumination, process variation, use, degradation, and mechanical positioning.

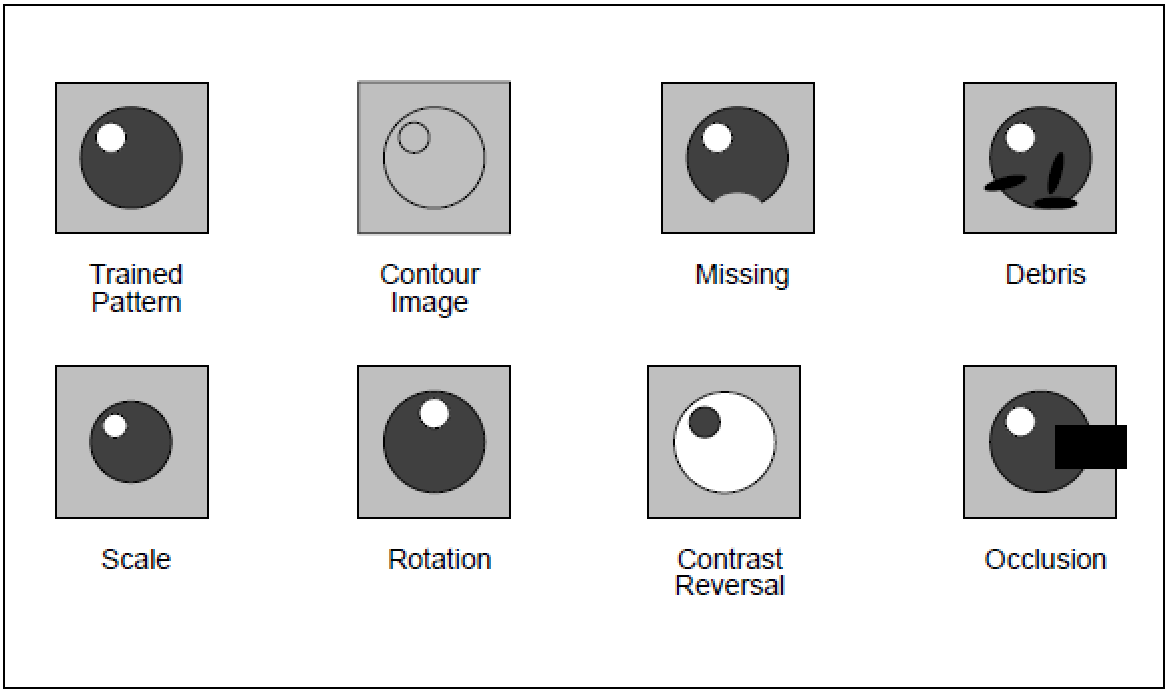


Figure 83. Pattern and Defects

Noisy Images

Some types of imaging devices, such as electron microscopes, produce images with a high noise level, and exhibit a non-linear reflectance behavior. For this type of image, Geometric Search is the better choice. However, with high-level noise the noise pixels could be picked up as real pattern pixels.

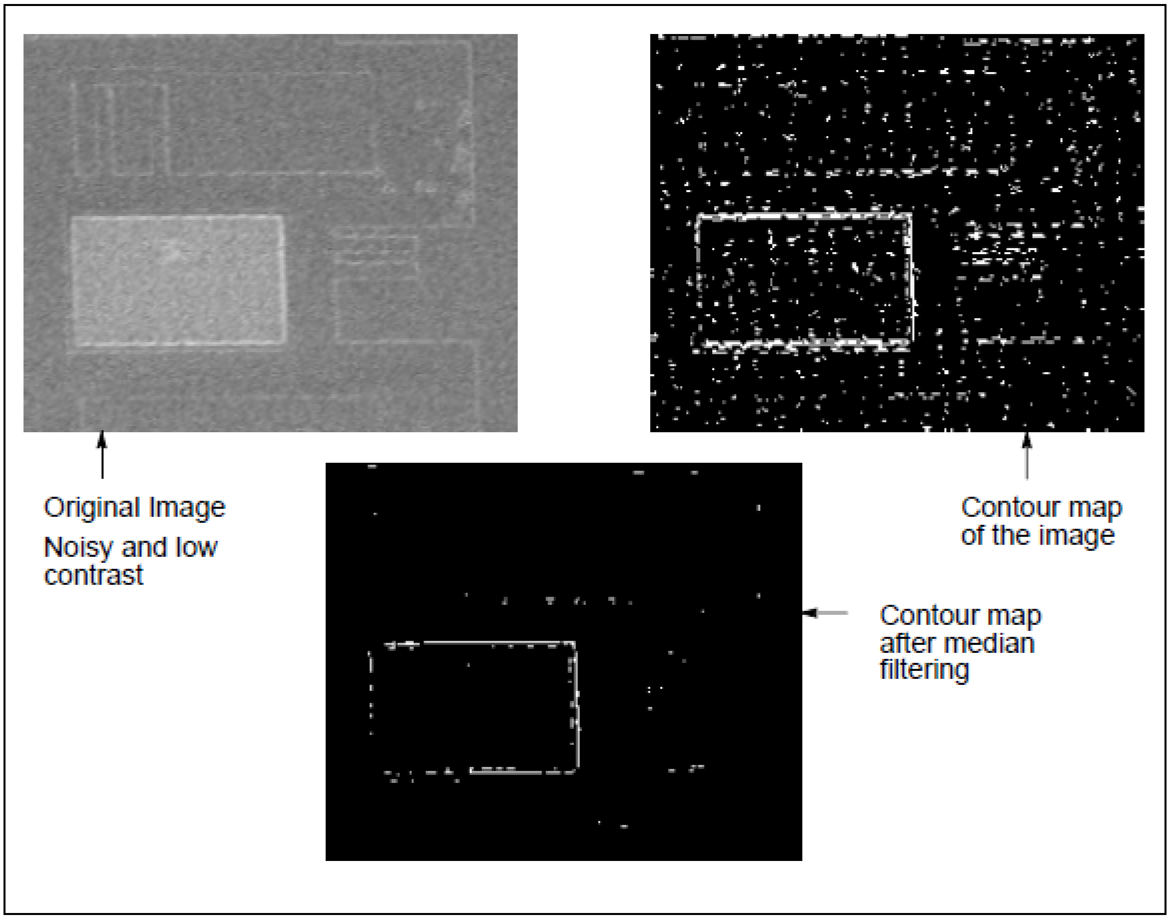


Figure 84. Noisy Images and Filtering

Contour Images

In Contour images, the pattern is defined only as a contour with about two pixels in thickness. The texture (filling) of the pattern in very similar to background; for example, a fiducial on glass with fine etching, or on a semiconductor wafer. Geometric Search operates based on the relationships between edge pixels (contour), and Quality measures the similarity between the trained pattern and a found instance of the pattern.

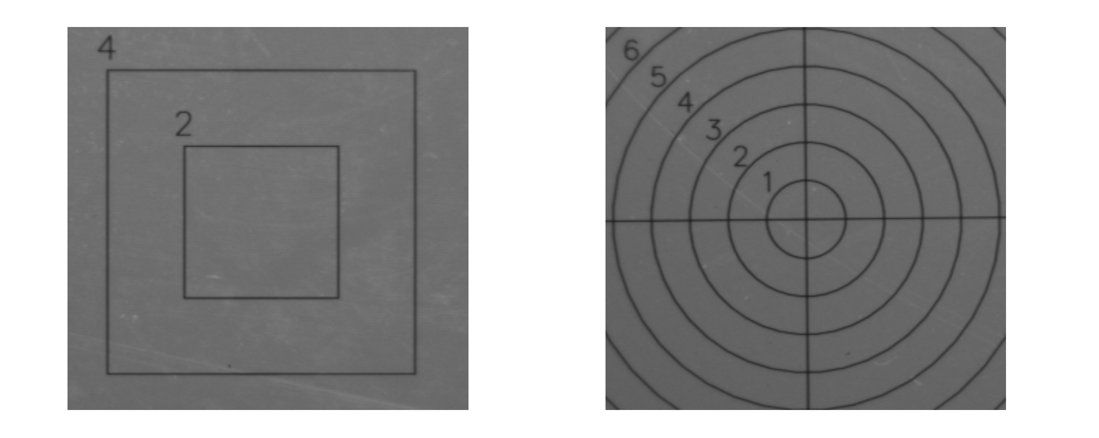


Figure 85. Contours

Contrast Reversal Images

Contrast Reversal can be caused by lighting on a smooth or polished surface. Contrast reversal is like having an image and the negative of the image.

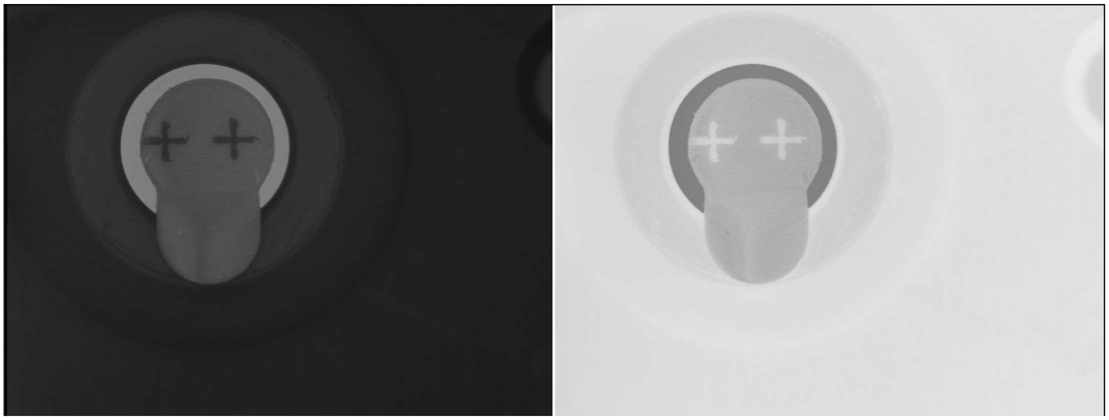


Figure 86. Contrast Reversal

Non-Linear Reflectance Images

Parts or objects with smooth or polished surfaces reflect light. Very slight changes in the angle or intensity of the light source results in very large changes in the gray-scale intensity levels. Under inspection conditions, light is often reflected unevenly (non-linearly). Dealing with non-linear reflectance has always been a problem in machine vision.

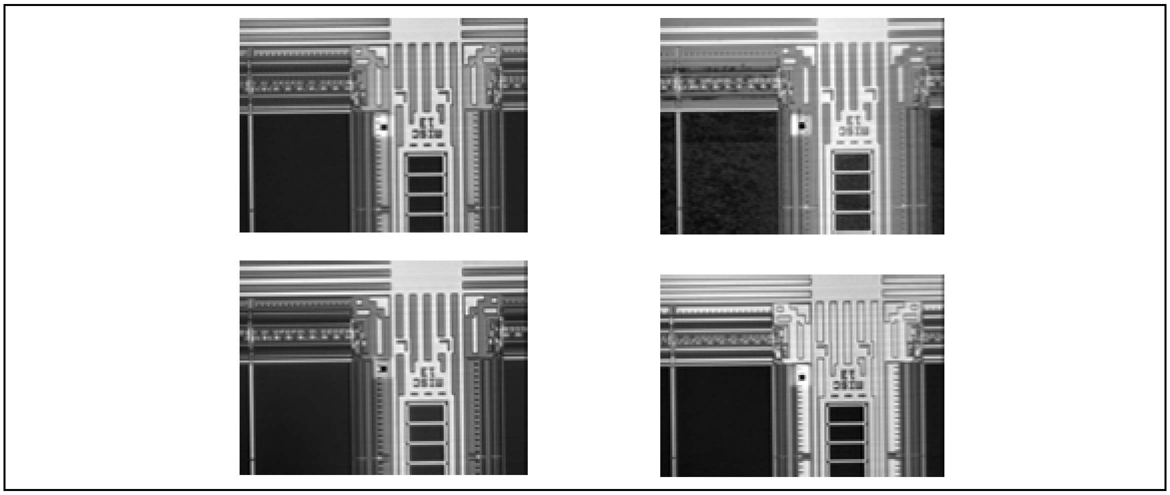


Figure 87. Non-Linear Images of Wafers

The four wafer images in Figure 87 have very different intensity values, but their edge maps are very similar, as shown in Figure 88. Geometric Search relies on the edge map, and is excellent at dealing with such images.

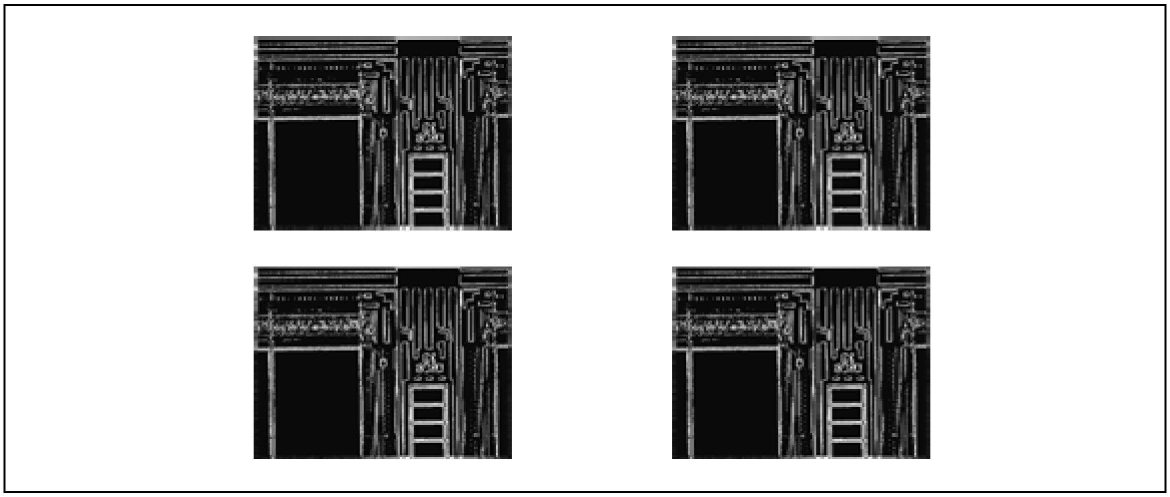


Figure 88. Edge Maps of Wafers

Figure 89 shows four wafer images inspected at much higher enlargement. Here the images have a large amount of information in the texture (intensity levels) of the image, however the edges in the image are blurry. Although Geometric Search would work for these images, Correlation Search makes a better choice.

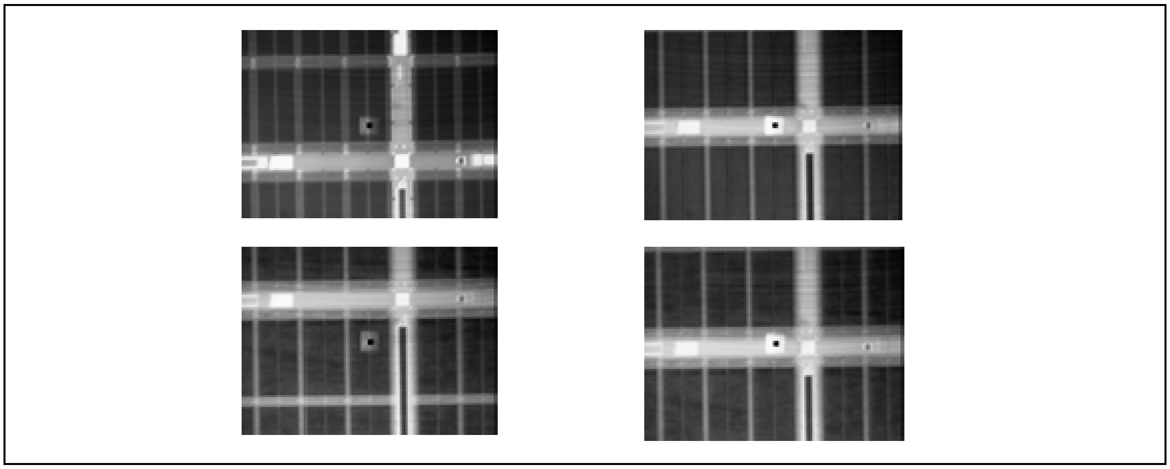


Figure 89. Wafers with Blurry Edges

When in doubt, it is always a good idea to quickly try both Correlation and Geometric Search and then select the approach best suited to the application.

Effects of Rotation

When searching without allowing rotation and the part or pattern actually is rotated more than a few fractions of a degree, the accuracy of Position, Quality, Conformance, values decreases, unless the pattern is rotational invariant (such as a circular fiducial). If the part is rotated, Search returns the higher accuracy when you specify that the pattern could be slightly rotated, but the speed is reduced. Enable rotation only when rotated parts are actually expected.

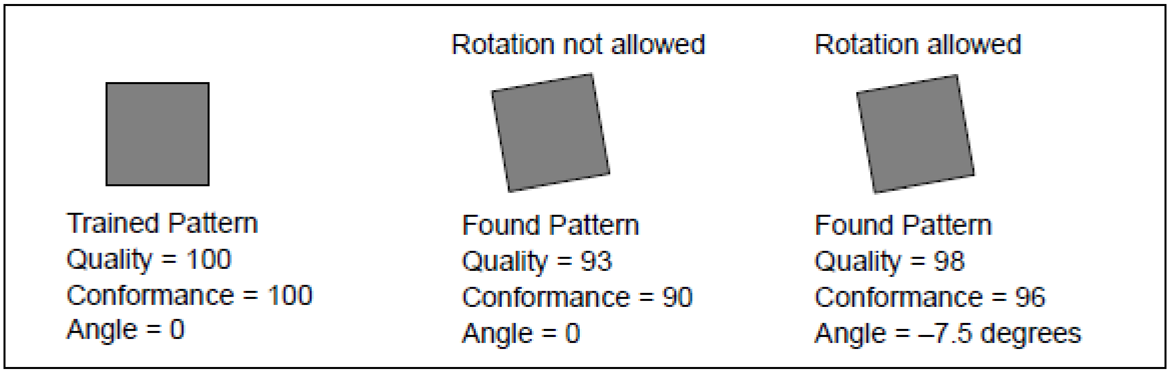


Figure 90. Rotation Effects

Aspect Ratio

Some cameras and frame grabbers can introduce an aspect-ration problem (aspect ratio not equal to 1). The cause of this is having rectangular pixels in the sensor with incorrect sampling rate of the frame grabber.

A circle appears as an ellipse when an aspect-ratio problem exists. Contours from a part would compress with a given orientation. This would result in reduction of robustness of the search.

Rotation Invariant Parts

Training on parts that have rotational invariant contours reduces the robustness of the search. A true circle does not produce any angle preference (any angle is good, the circle is the same at any angle) so it may influence the final decision. Also, if the circle was actually an ellipse, then search results may be influenced by the ellipse.

8. Inspect Algorithm

Inspect is a new generation self-teaching template-matching tool for image processing and object recognition. Inspect learns the range or variation of acceptable parts by “training” from several examples of good parts, and applies these rules to production parts.

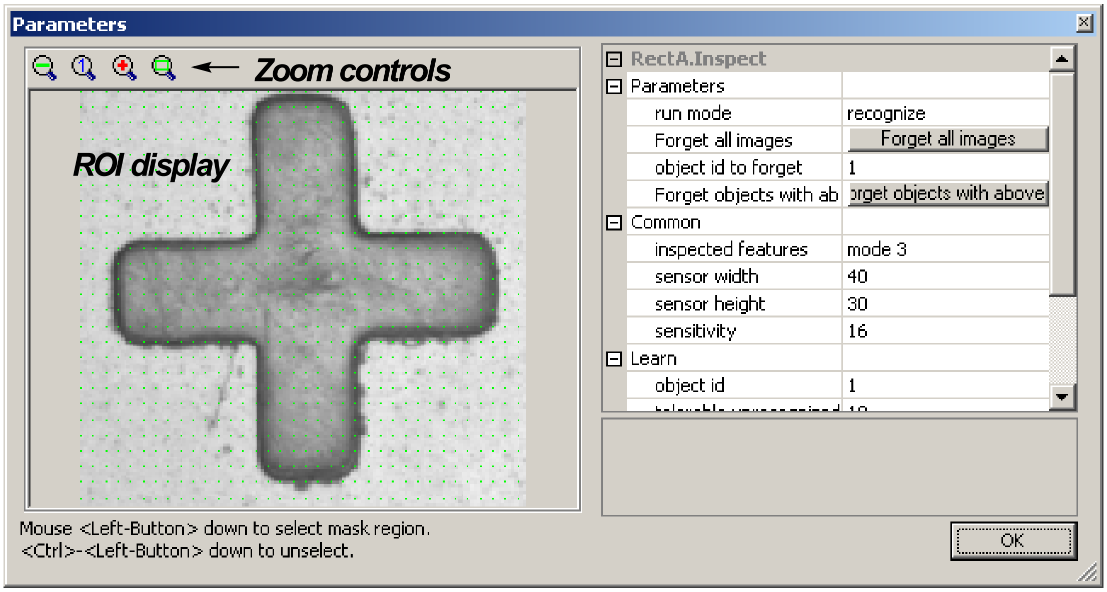
How Inspect Works

Inspect is based on gray-scale correlation template matching. The difference between “Golden Template” matching and Inspect, is that the Inspect algorithm learns more than one template image. Inspect trains on or learns from several images, to create a range of “good” or passing images. Inspect does not search the image window to locate the pattern. Do no t increase the ROI as you would for a search algorithm. You must align the Inspect ROI to find the correct pattern or part in the image window, if the part moves.

Inspect assigns an “Object ID” for identifying the learned or trained images. You can train more than one object ID, or more than one type of part or pattern to be recognized at runtime. This gives Inspect the ability to sort multiple objects, or sort objects according to specific trained defects, if you train common defects with separate object ID numbers. You can forget all images with a single Object ID number, or you can forget all images for all ID numbers.

Inspect Parameters

When you create a ROI and select the Inspect algorithm, the ROI is overlayed with a green grid. Click on the “Parameters” button (in the ROI Options ➔ Processing dialog), to open the custom Inspect training tool. The ROI image is displayed in the left pane. This display can be zoomed with the buttons at the top of the pane. The (green) dots are the intersections of the grid displayed in the Image Window. You can create “ignored areas” in the ROI by selecting rectangles in the grid. Left-click on the grid. A rectangle appears. This area is ignored at run time. Hold the Control key to remove ignored areas.



RectA.Inspect	
Parameters	
run mode	recognize
Forget all images	Forget all images
object id to forget	1
Forget objects with ab	orget objects with above
Common	
inspected features	mode 3
sensor width	40
sensor height	30
sensitivity	16
Learn	
object id	1

Mouse <Left-Button> down to select mask region.
<Ctrl>-<Left-Button> down to unselect.

Figure 101. Inspect Parameters and Image

NOTE *Inspect does not automatically enter Learn mode when you open this window. Inspect does not automatically enter Run mode when you close this window. You must take care to set the correct mode for Learn and Run before you close this window.*

Training Images

1. Change the Run mode to “Learn”.
2. Close the training tool, and Close the ROI Properties.
3. Run the investigation once or run continuously with an external trigger.
   Use good or perfect parts or images for the training process.
4. Halt the investigation.
5. Reopen the Inspect tool.
6. Change the Run mode to “Recognize” to test or inspect parts.

Don’t Care Area Editing

The small dots shown in the ROI image (in figure 100) create a grid. You can use this grid to mark out areas to ignore.

Select Mode - As you click or drag your mouse, rectangles appear, showing the areas that will be ignored.

Erase Mode - Hold the Control key as you click or drag your mouse over the marked areas, the rectangles disappear.

A. Accelerator "short Cut" Keys

Use Accelerator Keys to access commands. Sherlock provides multiple access methods for many functions.

NOTE *It is possible to set up infinite loops in Sherlock, either by waiting forever for I/O conditions, or by jumping to the same Subroutine Instruction or Image Window over and over. ESC terminates these "infinite wait" conditions. Press and hold ESC for a few seconds to clear looping conditions.*

Table A-1. Accelerator Key Table

Keys	**Equivalent To:**
Esc	Stop after completing investigation loop
Shift-Esc	Stop immediately
F1	Help
F8	Investigate continuously
F9	Investigate once
F10	Execute single step
F11	Run Calibration subroutines
Ctrl-C	Copy selected instruction(s)
Ctrl-L	Lock or Unlock user interface
Ctrl-N	New investigation file
Ctrl-O	Open an investigation file
Ctrl-S	Save investigation file
Ctrl-V	Paste (tree)
Crtl-X	Cut or Delete selected instruction(s)
Crtl-Z	Undo paste
Ctrl-Insert	Copy selected instruction(s)
-	Zoom out (minus key)
=	Zoom in (equal key)
Shift	increases speed in Highlight mode
Shift-Delete	Cut or Delete selected instruction(s)
Shift-Insert	Paste
Ctrl	turn off stickiness when moving an undocked window

B. Annotation Colors

Annotations, and other graphic options, let you define your own colors for on-screen display:

- "color" [Double] - a decimal (or base 10) number for a B-G-R (Blue-Green-Red) color value. You can enter a hex number (e.g. 0xFF0000). It will be converted to base 10.
  Default value: 0.00 [0.00000000000000000] color black.

You can define up to "16 million" (16,777,215) different colors. You can enter the color value as a decimal number, or as a hexadecimal number. You do not need to enter leading zeroes for hexadecimal numbers. Hexadecimal values are converted and stored in binary, and displayed in decimal.

The following table gives 21 of the more common or popular values. You can also use any of the color palette tools in Windows (Display Properties, Paint, Word) to find B-G-R values for other colors, and convert to hexadecimal.

Color	*Hexadecimal*	*Decimal*	*Blue*	*Green*	*Red*
Black	0x0	0	0	0	0
Dark gray	0x808080	8421504	128	128	128
Light gray	0xC0C0C0	12632256	192	192	192
White	0xFFFFFF	16777215	255	255	255
Bright red	0xFF	255	0	0	255
Bright yellow	0xFFFF	65535	0	255	255
Bright green	0xFF00	65280	0	255	0
Aqua	0xFFFF00	16776960	255	255	0
Bright blue	0xFF0000	16711680	255	0	0
Magenta/Fuchsia	0xFF00FF	16711935	255	0	255
Light yellow	0x80FFFF	8454143	128	255	255
Light green	0x80FF80	8454016	128	255	128
Medium blue	0xFF8080	16744576	255	128	128
Medium red	0x8000FF	8388863	128	0	255
Orange	0x80FF	33023	0	128	255
Dark purple	0x800080	8388736	128	0	128
Dark green	0x8000	32768	0	128	0
Dark blue	0xC08080	12615808	192	128	128
Navy blue	0x800000	8388608	128	0	0
Dark yellow	0x8080	32896	0	128	128
Brown	0x4040	4210816	64	64	128

Glossary

This section contains descriptions of some Sherlock terms and features.

Algorithm - Algorithms take an image as input, extract information from the input image, and output data or "reading". A preprocessor takes an image as input, performs some operation on the image data, and returns a different output image.

Bitmap Image - All preprocessors share 10 pixel buffers (bitmaps). These are used primarily for the logical AND, OR, XOR instructions. Bitmaps exist only in memory and thus are lost when exiting Sherlock. When Sherlock performs an operation on a ROI and bitmap of different sizes, only pixels within the minimum intersection of the ROI and bitmap sizes are affected.

Correlation Search - Searches for matches based on gray-scale values (texture or filling) in the image or ROI and scores the match from 0 to 100%.

Conformance - Percentage of the trained pattern area that was found (within the conformance tolerance) in the instance of the pattern.

Contour - A pattern that is defined by shape (rather than by pixel intensity values).

Connectivity - Another name for a "Blob Tool". Locates objects or "blobs" in an image (usually in a binary image) and counts or measures these objects.

Dongle - A software key or license key, usually attached to a USB port, or the parallel port or printer port of a personal computer.

Edge - The boundary between two regions in an image that have different pixel intensities.

Edge Map - The edge map of an image is an image which only has the edges of the image.

Edge Score - Similarity between the edges of the trained pattern and the found instance.

Edge Sharpness - The rate of change (slope) in the pixel intensities on the two sides of an edge.

Edge Strength - The magnitude of change (difference or threshold) in the pixel intensities on the two sides of an edge, used to qualify it as an edge in the edge map. Edge strength is the range of intensity change.

Filter Size - Minimum number of connected pixels to define an edge. Groups smaller than the Filter Size are ignored in defining the Contour of the Trained Pattern.

Flow Instruction - A flow instruction causes the Sherlock program focus to jump to another instruction in the subroutine, to call another subroutine, or to return from the current subroutine.

Found Pattern - An instance of the Trained Pattern that is found in the search image.

Frame Grabber - Add-on card or circuit board for a PC, to allow acquisition of digitized video images.

Function - A function is a mathematical or geometric operation performed in a formula.

Image Analysis - Image analysis is an operation which examines a picture and generates some form of data from analyzing the picture.

Image Preprocessing - Image preprocessing is any operation which is designed to improve picture quality; these operations, such as thresholding, smoothing, morphology, or other image filtering, turn one picture into another picture.

Image Tag - The Image Tag is an index number, incremented with each (learned) image.

Image Window - A image window is a virtual camera which snaps a picture (and retrieves a selected portion of the picture) for processing.

Instruction - A subroutine instruction performs a mathematical or geometric function on input readings and variables and produces output reading(s). An I/O control instruction generates, tests, or waits for a particular input/output condition. These instructions may be tied to serial, parallel, analog, or other ports.

Investigation - The Sherlock inspection program; a collection of subroutines, each of which contains instructions. An investigation is like a computer program, except that it is created entirely in a graphical, point-and-click environment.

Model - Representation of a pattern, stored as an image and/or data that represents that image

Monitor Window - The monitor window displays benchmark timings and Sherlock internal error messages.

Object ID - Inspect can be trained to sort or distinguish between different types of objects. Each different type is assigned a Object ID, to identify it. This could be different packaging, different grades of a part, completely different parts, completely unrelated product lines (the ROI must be the same in all cases).

Preprocessor - A preprocessor takes an image as input, performs some operation on the image data, and returns a different output image.

Program - Program is a sequence of subroutines and instructions that provide a flowchart of a given machine vision investigation.

Quality - Strength of match of the gray-scale levels in the conforming areas of the instance.

Reading - A reading is a value returned by an algorithm or an instruction. Readings have the following types: number (N), point (P), Boolean (B, either true or false), line (L), or Character String (S). Each reading type can also appear in an array, for example N[ ] is an array of number readings. Sherlock uses readings to generate reports, make pass/fail decisions, influence execution flow, or as input to a formula.

Report Window - The report window displays user-defined messages.

ROI - The ROI (region of interest) is the basic image analysis instruction. Several types of ROIs (lines, rectangles, points, and circles) are created with the mouse during setup and have individual image processing and analysis operations associated with them.

Rotation - Most of the Sherlock application measures rotation or angle in milli-radians, or 1/1000 radians, except where noted otherwise. A value of 1570 represents 1.570 radians. Unless specifically converting to degrees or radians, the rotation is represented in milli-radians.

SPC - Statistical Process Control. Sherlock gathers some data that may be used or analyzed by external SPC software programs.

Score - Score is a measure of the similarity between the entire trained pattern and the entire found pattern. Score is computed by subtracting the trained pattern from the instance and then computing the variance. The score is the variance scaled from 0 to 100, with 0 being the variance of the trained pattern with a blank image, and 100 being the variance of the trained pattern with an identical image.

Search Image - An image where you are looking for the search pattern (or trained pattern).

Search Pattern - Reference to the Trained Pattern during the search process. (also: "Search Trained Pattern")

Subroutine - A subroutine is a sequence of machine vision instructions.

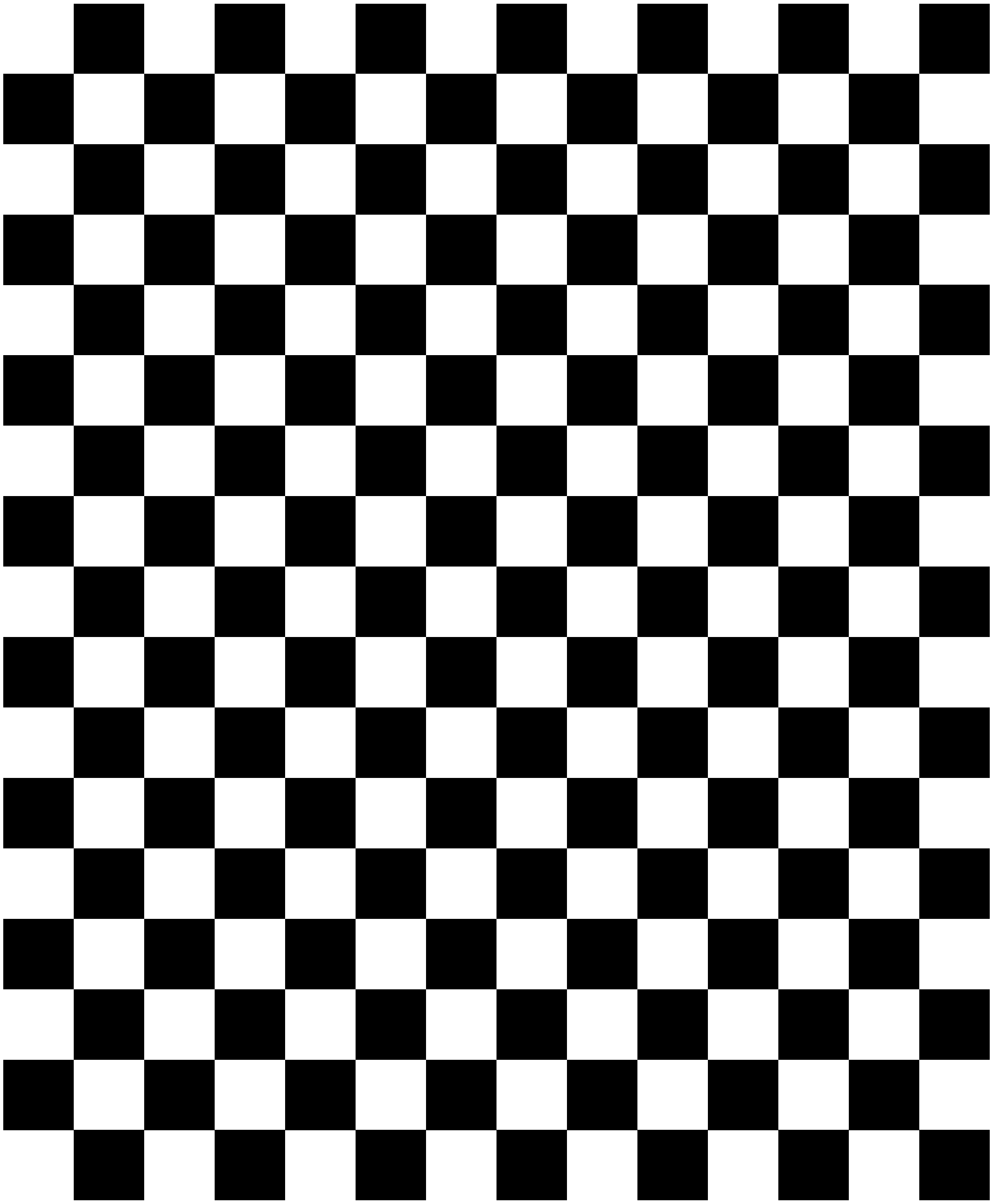
Texture - A pattern that is defined by pixel gray-scale intensity values.

Trained Pattern - Image of the pattern (or part) that you trained (or learned).

Variable - A variable is storage element used for maintaining counts, inspection states, or to allow the passing of dynamic parameters. Sherlock displays variables in the Variables window next to icons showing their type (N, P, and so on).

Watch Window - A watch window is a screen location for display of readings.

Calibration Checkerboard



Calibration Checkerboard - 1 cm or 10 mm or 0.40 inch Squares